Zemax OpticStudio 14.2 SP1

OpticStudio Help Files

October 10th, 2014

Contents

Using Glass Catalogs 187

Polarization Analysis 209

Optimization Overview   237

Thermal Analysis of Optical Systems   379

Physical Optics Propagation   387

Sequential Surfaces   411

Non-Sequential Overview   497

Non-Sequential Geometry Objects   543

Non-Sequential Sources 631

Non-Sequential Detectors 649

Tolerancing 661

The Analyze Tab

The Analyze Tab (Non-Sequential UI Mode)   1171

The Optimize Tab   1223

The Optimize Tab (Non-Sequential UI Mode) 1249

The Tolerance Tab 1261

The Libraries Tab 1283

Part Designer Tab   1339

The Programming Tab   1367

Getting Started With Zemax OpticStudio™ 14.2 SP1

Congratulations on your purchase of Zemax OpticStudio!

OpticStudio is the industry standard optical system design software, combining sequential lens design, analysis, optimization, tolerancing, physical optics, non-sequential optical system design, polarization, thin-film modeling and mechanical CAD Import/Export in a single, easy-to-use package.

Although OpticStudio is easy to use, optical system design is a very broad area of engineering. This guide is intended to get you started using OpticStudio quickly. It is the first place to start if you are new to OpticStudio, or if you are upgrading from previous versions of Zemax.

We strongly recommend you take the time to work all the way through this infromation on getting started with OpticStudio. It covers:

- Installing OpticStudio, and customizing its appearance and file locations to your preference
- Entering a simple sequential design
- Understanding the normalized definitions OpticStudio uses
- An overview of the multiple configurations capability
- How to export components and rays to mechanical CAD packages
- Optimizing a simple lens
- Using some of the powerful tools OpticStudio makes available
- Tilting and decentering optical components
- Entering a simple non-sequential system, tracing rays, and using detectors
- Colorimetry
- Thin-Film Coatings
- Surface, bulk and fluorescent scattering

Also, our web-based Knowledge Base at www.zemax.com/kb is an indispensible resource for all OpticStudio users. It contains tutorials, worked examples and answers to many frequently-asked questions.

Important notice

Zemax® is a registered trademark of Zemax LLC Copyright © Zemax LLC 1990-2014. All rights reserved.

OpticStudio™ is a trademark of Zemax LLC 2014. All rights reserved.

ReverseRadiance and LightningTrace are trademarks of Zemax LLC Copyright © Zemax LLC 1990-2014. All rights reserved.

SolidWorks® is a registered trademark of Dassault Systèmes SolidWorks Corporation.

AutoDesk Inventor® is a registered trademark of Autodesk, Inc.

Creo Parametric® is a registered trademark of Parametric Technology Corporation.

MATLAB® is a registered trademark of the The Mathworks, Inc.

All other product names or trademarks are property of their respective owners.

Information in the help file documentation is subject to change without notice and does not represent a commitment on the part of the vendor. The software described in this documentation is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement.

Zemax LLC provides this publication "as is" without warranty of any kind, either express or implied, including but not limited to the implied warranties or conditions of merchantability or fitness for a particular purpose. In no event shall Zemax LLC be liable for any loss of profits, loss of business, loss of use or data, interruption of business, or for indirect, special, incidental, or consequential damages of any kind, even if Zemax LLC has been advised of the possibility of such damages arising from any defect or error in this publication or in the Software.

Installation

There are two simple steps to installing OpticStudio, which must be followed in order. All files are downloaded from www.zemax.com/downloads.

Step 1: Install the Key Driver and Plug in the Key

The key driver installation is straightforward. Double-click the key driver installer once you have downloaded it, and choose the 'Complete' installation of all program features.

A dialog box will also ask for your permission to modify the firewall settings of your computer to allow remote users of your computer to run ZEMAX using Remote Desktop. If you want to authorize this, click "Yes", otherwise click "No". To change this setting, just re-run the key driver installer.

Plug the key in once the key driver installation is complete, and Windows will detect the hardware key. The green LED at the end of the key will illuminate.

Step 2: Install Zemax OpticStudio

The OpticStudio installer is similarly straightforward. Download and run the installer and step through the on-screen instructions. You may choose where within the \program files\ hierarchy OpticStudio is installer (by default this is C:\Program Files\Zemax OpticStudio\). If you choose a different folder, any future updates you install will remember this location and treat it as the default, until you define a different location.

Note that the installer will automatically scan your computer to verify the presence of Microsoft prerequisites that are required to run OpticStudio. If any of the prerequisite components are missing, the necessary files will be downloaded and installed. These will subsequently be kept up to date by Windows Update.

For more detailed instructions on installing OpticStudio for the first time, please our knowledge base article:
http://www.zemax.com/support/knowledgebase/installing-opticstudio-for-the-first-time

License Codes

OpticStudio contains built-in license codes for all keys that were eligible to run this version at the time this version was compiled. If you have purchased a new key, this license code will not be built in, and OpticStudio will download the latest version of the license code file from the zemax.com website automatically. If for any reason it still cannot get a valid license code, the following dialog will appear:
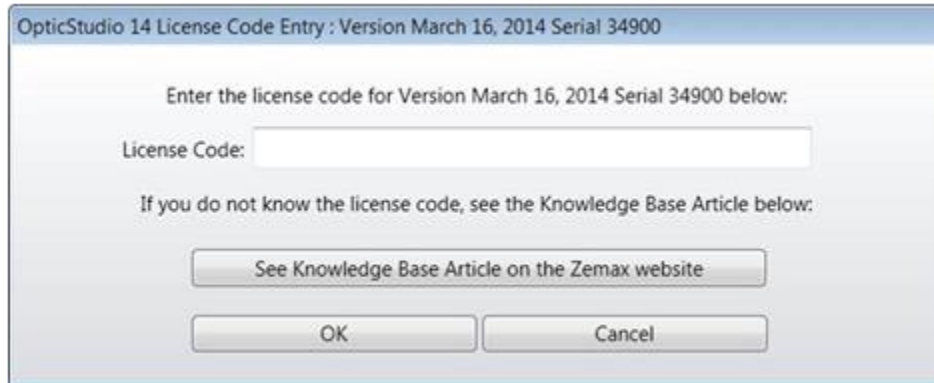
Use Alt-Print_Screen to copy the dialog to the clipboard and paste it into an email to support@zemax.com. We will promptly send you the license code or further instructions.

Note: Please do not phone for a license code! License codes are complex multi-character strings and cannot be reliably given over the phone. Emailing the screenshot of the dialog box to us is the quickest and most error-free way of getting your license code.

Network Keys and Clients

Zemax OpticStudio can also be supplied with 5, 10, 25 and 50-user network keys. Installation is almost identical, except that the key driver and hardware key are installed on one computer (called the 'keyserver' machine) and Zemax OpticStudioand the Prerequisites installed on as many other machines as you wish (the 'client' computers). When a client machine starts OpticStudio, it looks to the keyserver machine to see if a license is available, and if so, OpticStudio starts.

Installation of the key driver on the keyserver machine is identical to the normal installation, except that you obviously MUST allow the firewall settings to be adjusted to allow network access to the key.

Installation of OpticStudio on the client machines is also identical to the normal installation, except that you must tell OpticStudio where to look for the keyserver machine after installation. Navigate to whatever folder you installed OpticStudio in (by default this is C:\Program Files\Zemax OpticStudio\) and locate a file called sntlconfig.xml.bak. Rename this file to sntlconfig.xml, and open it in Notepad. Edit the following line:

```
<ContactServer>
10.0.0.1
</ContactServer>
```

Replace the default entry 10.0.0.1 with the IP address of your keyserver machine and save the file.

Troubleshooting

OpticStudio will run without problem in the vast majority of cases. If you do experience problems, then visit our Knowledge Base at www.zemax.com/kb. Look at the Category 'Installation and Troubleshooting' for help.

Make sure your key is plugged in!

Customizing Your Installation

OpticStudio installs using several default settings which you may prefer to change. You can also set these defaults up on a project basis, so that all designs for a specific project use the same settings. Start OpticStudio, and navigate to Setup…Project Preferences:
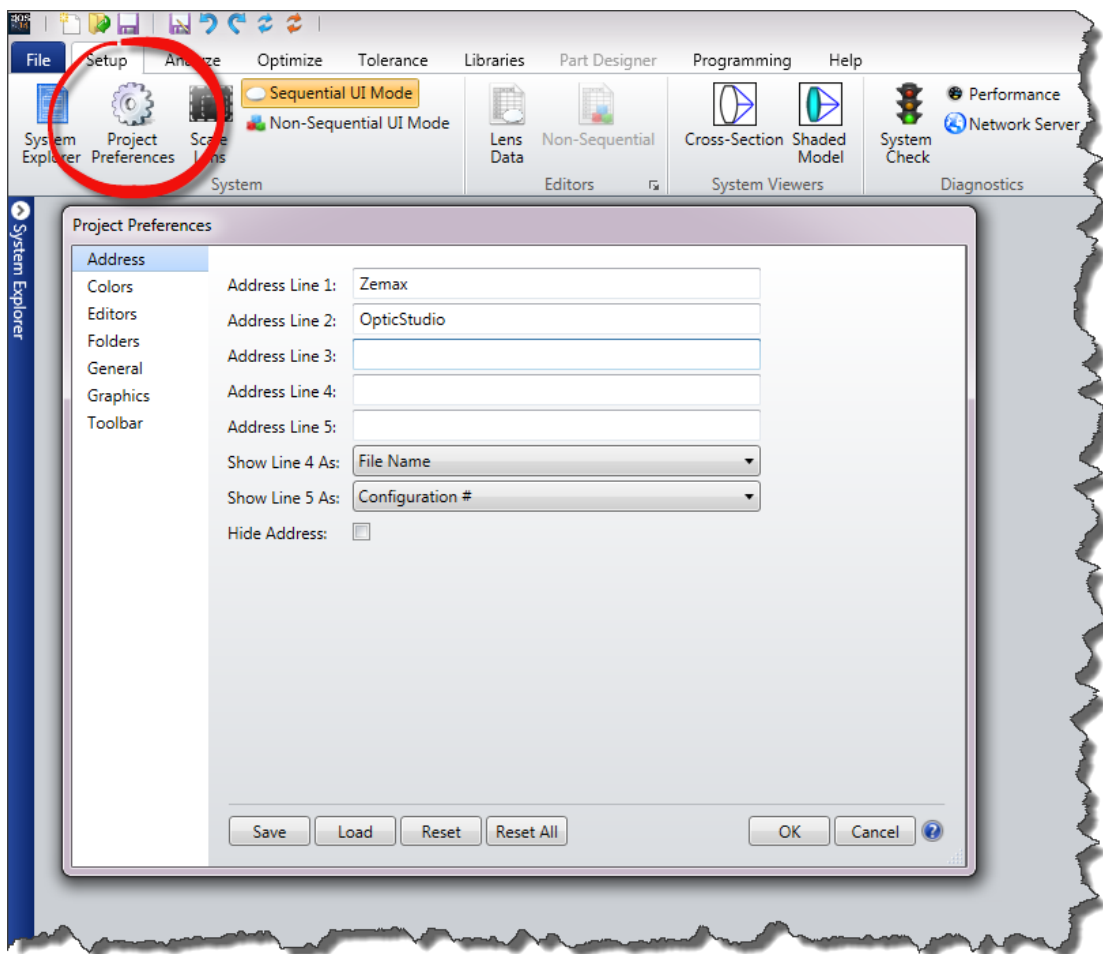
Navigating the OpticStudio Interface

The OpticStudio user interface provides a fast, flexible platform for the design and evaluation of almost any optical system.
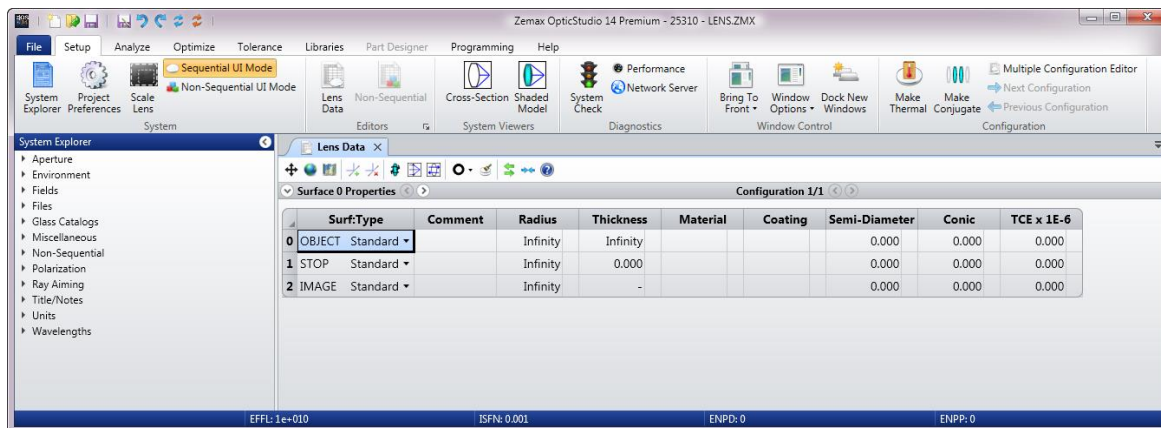
It consists of several key areas:

1. A **ribbon bar** which provides easy access to all the program's features, clustered in task-oriented tabs. Each tab contains groups of icons. In this example:
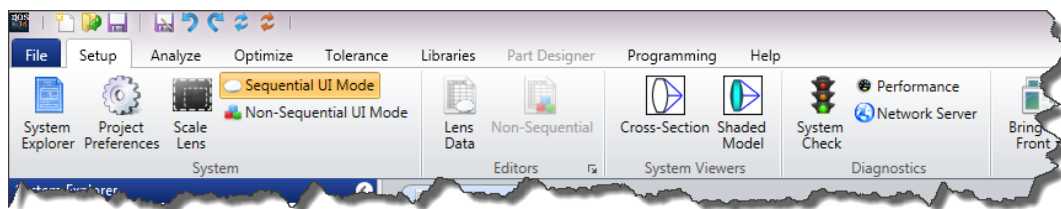
Setup is the name of the ribbon bar, and *System*, *Editors*, and *System Viewers* are examples of groups.

2. The System Explorer, which can be displayed or hidden at any time. This contains system-specific information about the optical system under design.
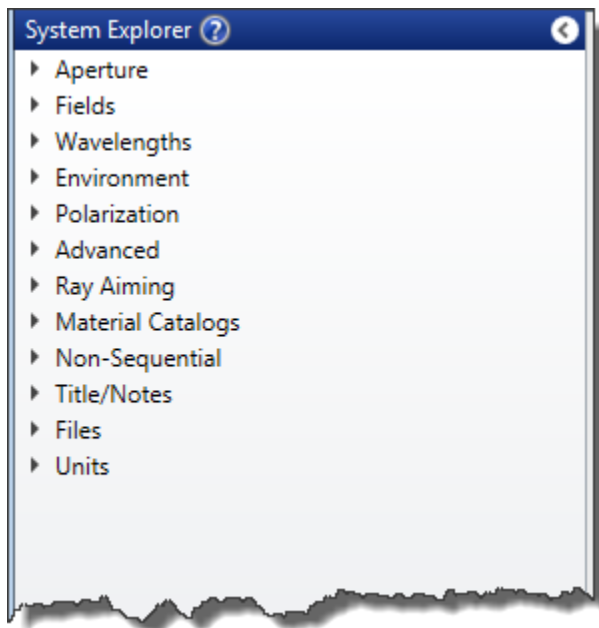
3. A **workspace**, which is the main area in which you will do your work.

4. A user definable **Quick Access Toolbar** which allows you to place your most commonly used features on the desktop for single-click access.

The tool bar is configured under Setup…Project Preferences…Toolbars.
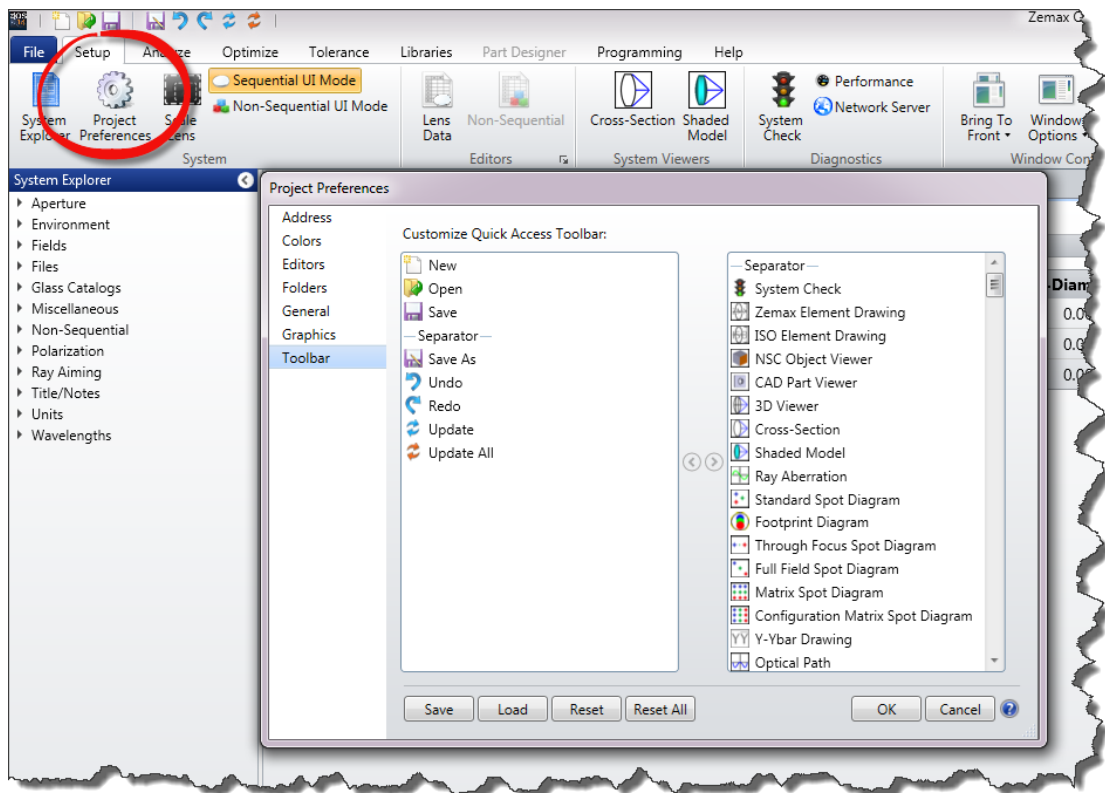
Multiple Project Preferences may be created and recalled to customize the user interface to the work in hand.

5. A **status bar**, which displays useful information about the design at the bottom of the workspace

System Explorer
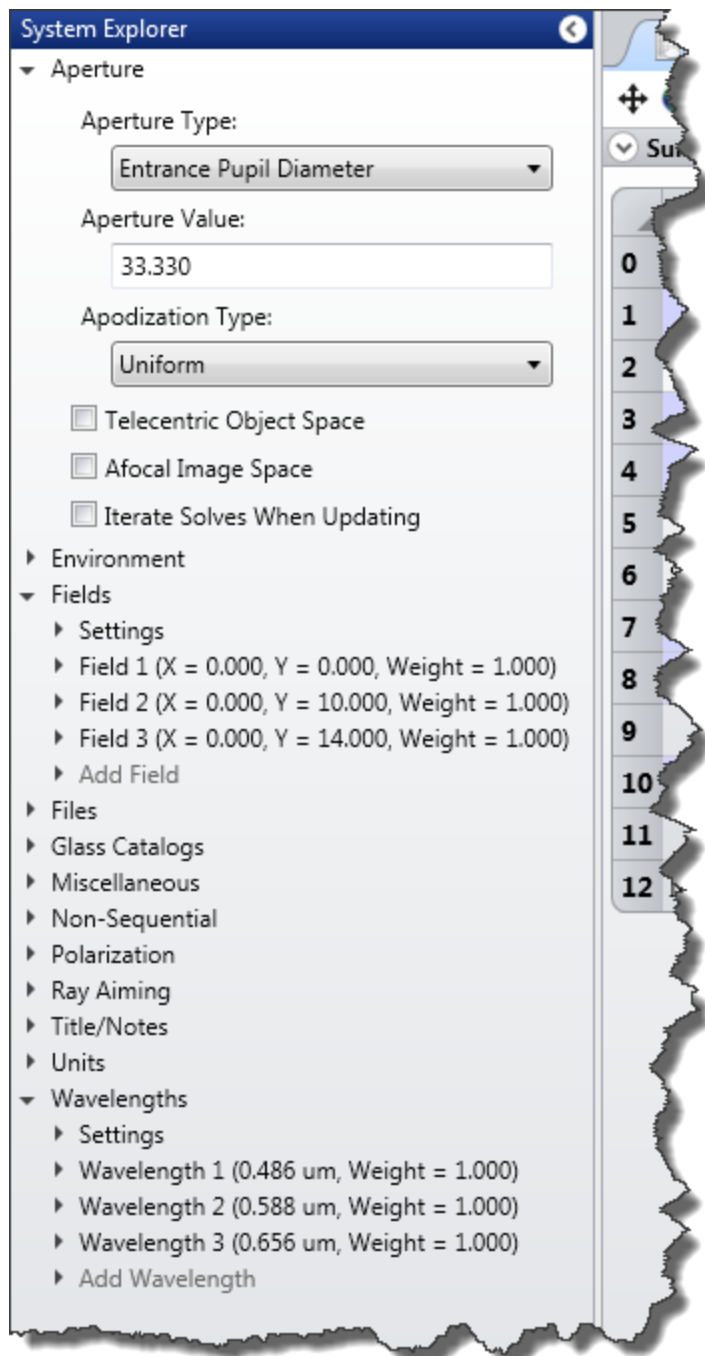
The System Explorer shows the basic system settings that are usually input at the very start of a design project. It is not usual to optimize these parameters, but it can be done if needed.

The groups within the System Explorer can be expanded, closed, and reorganized as desired, and the whole System Explorer can be hidden or expanded at any time using the < or > controls

Full details are given in the [System Explorer](#) section of The Setup Tab chapter.

File Tab
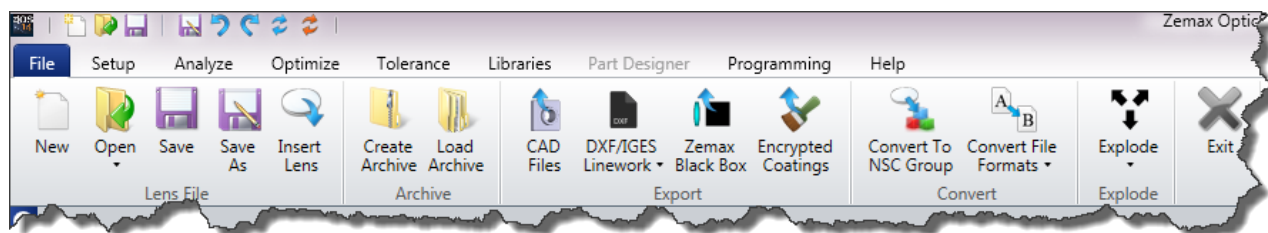
The File tab contains all the file input/output functions, split into these groups:

The **Lens File** group contains all the normal Windows file management tasks like file opening, saving etc. OpticStudio files are stored in .ZMX format files, along with associated .CFG files that contain configuration settings and .SES files that contain settings data for all windows that were open at the time the file was saved.

The **Archive** tab lets you create and open OpticStudio Archive files. These are stored in .ZAR format, and contain all the files needed to open the file on another computer with OpticStudio installed on it. All OpticStudio data, glass catalogs, coatings, CAD files, SolidWorks™ etc files used by the lens design are compressed into this single file so that you can easily make backups of the design as it progresses through the design process, or transfer the design to another machine.

The **Export** tab provides access to all the export capabilities in OpticStudio, including exporting to STEP, IGES, SAT and STL format CAD files, and exporting to DXF and IGES linework diagrams.
The OpticStudio Black Box feature allows you to encrypt a range of surfaces in the lens data spreadsheet, which can be given to other OpticStudio users without revealing the details of the design itself. This lets you give fully ray-traceable files to customers, or to other colleagues on a need-to-know basis, that give exact results on ray tyrace but do not reveal the design prescription.

The **Encrypted Coatings** feature is a similar capability for thin-film coatings, and allows the full prescription of a thin-film coating to be exported in an encrypted format that allows exact ray tracing without making the design itself available.

The **Convert** tab lets you convert between OpticStudio' sequential (lens design) and non-sequential (system design) modes, and also converts various file formats (like .MAT Matlab® , .INT interferometer data and .f3d OptiWave data) between original and OpticStudio formats.

See the File Tab chapter for full details of all options.

Setup Tab
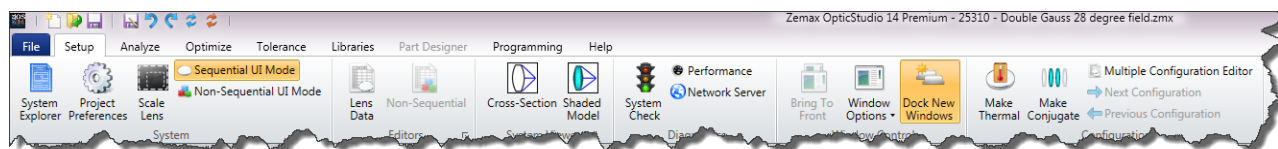
This tab is used when you start each design project, and is visited infrequently or not at all after initial setup.

The **System** group puts everything to do with basic system setup in one place. Project Preferences lets you customize the OpticStudio installation, folder locations, quick access toolbar etc and save these settings to a project configuration file.

The **Editors** group gives access to the spreadsheets that are used to define the optical system on a surface by surface or object by object basis.

The **System Viewers** tab gives access to the layout plots that are used to view the optical system itself.

Errors and Tests lets you check the OpticStudio file. The System Check utility traps many common setup errors.

Windows Control defines how windows in the OpticStudio workspace behave. Windows can be docked, float freely, be tiled and cascaded etc.

Configuration is used is the design has multiple versions, which is typically the case for zoom lenses, thermal analysis of lenses over a range of temperatures, scanning lenses and lenses with moving parts. The Configuration group appears on all ribbons if more than one configuration is defined.

See the Setup Tab chapter for full details of all options.

Analyze Tab
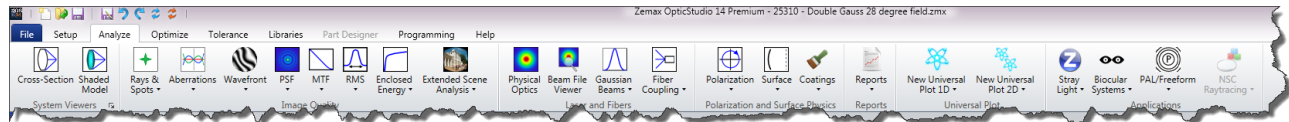
This tab provides access to all of OpticStudio' analysis features in sequential UI mode. Almost all imaging system design is done in this UI mode. Analysis features provide detailed performance data over a wide range of requirements. Analysis features never change the underlying design, but provide diagnostic information on the design that is used to guide any required changes.

The **System Viewers** tab gives access to the layout plots that are used to view the optical system itself.

The **Image Quality** tab provides access to all the analyses used in the design of imaging and afocal systems, including ray-trace data, aberration data, wavefront, Point Spread Functions and more.

Lasers and Fibers gives access to laser-system-specific features such as simple Gaussian Beam Analysis, Physical Optics and fiber coupling calculations.

Polarization and Surface Physics computes the performance of thin-film coatings on individual surfaces, overall system performance as a function of polarization and plots of surface sag, phase and curvature.

Reports provides text based analyses for presentation purposes.

Universal Plots allow you to create your own analysis features if required.

Applications show application-specific analysis features such as Biocular system analysis, freeform and Progressive Addition Lens analysis, and also access to the full non-sequential features of OpticStudio.

If the lens uses multiple configurations, the Configuration group (see Setup tab) is also shown.

See the Analyze Tab (Lens Design Mode) chapter for full details.

Analyze Tab (Non-Sequential UI Mode)
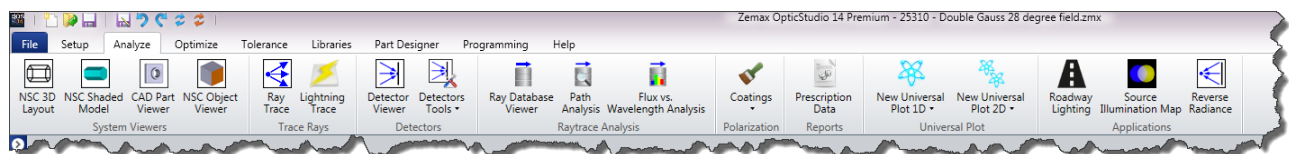
This tab provides access to all of OpticStudio' analysis features in non-sequential UI mode, which is used for most illumination, stray light, lighting, etc. designs. Analysis features provide detailed performance data over a wide range of requirements. Analysis features never change the underlying design, but provide diagnostic information on the design that is used to guide any required changes.

The **System Viewers** tab gives access to the layout plots that are used to view the optical system itself.

Trace Rays initiates ray tracing using either the comprehensive non-sequential ray-trace engine, or a faster, approximate method called LightningTrace™ that is very useful if the source can be approximated as a point source.

Raytrace Analysis provides a wide range of analyses on a previously performed raytrace.

Polarization computes the performance of thin-film coatings on individual faces of objects.

Universal Plots allow you to create your own analysis features if required.

Applications show application-specific analysis features such as Roadway Lighting Analysis

See the Analyze Tab (Non-Sequential Mode) chapter for full details

Optimize Tab
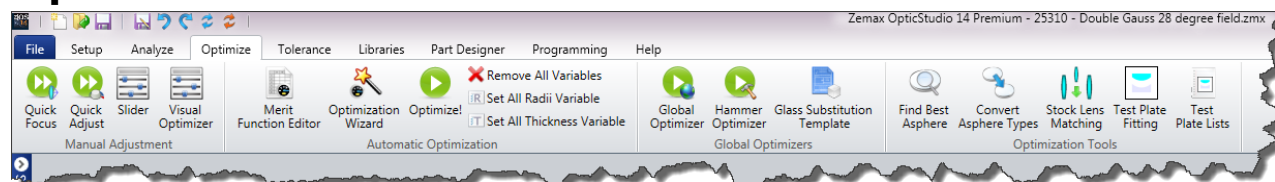
This tab is where you control the Zemax optimization capabilities.

Manual Adjustment contains a series of features that let you manually adjust the design for desired performance. This group is only available in Sequential UI mode.

Automatic Optimization provides access to the Merit Function Editor, which is how a system's performance specification is defined in Zemax. The Optimization Wizard lets you quickly generate a merit function based on the most common requirements (smallest spot, best wavefront error, least angular deviation) which can then be edited to your exact requirements.

Global Optimization is used in two main scenarios: at the start of a design process (Global) in order to generate design forms for further analysis, and after initial optimization (Hammer) to exhaustively improve the current design.

Optimization Tools performs a range of post-optimization functions such as finding the best surface to aspherize or swapping lenses in the current design for catalog optics. This group is only available in Sequential UI mode

See the Optimize Tab and Optimization Overview chapters for full details.

Tolerance Tab
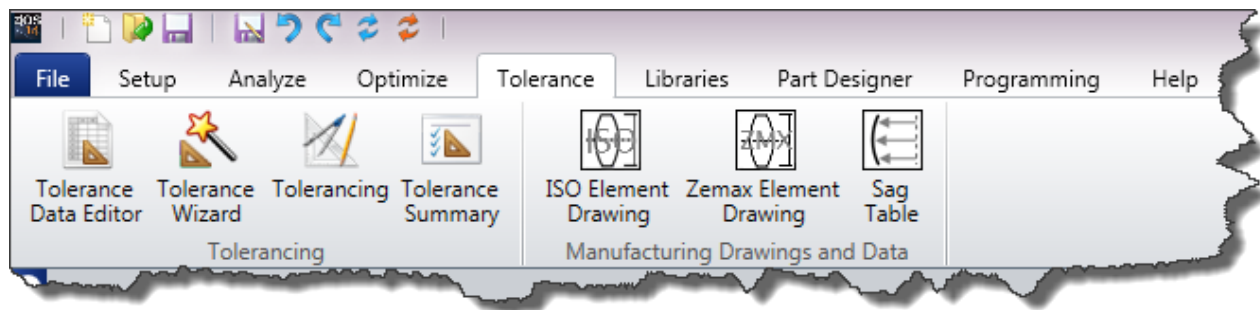
Tolerancing allows you to see the effect of finite manufacturing and assembly tolerances on the design.

In the Tolerances group, the Tolerance Data Editor is where you will enter the tolerances you wish to place on each parameter. The Tolerance Wizard will quickly set up a set of tolerances which you can then edit.

The **Manufacturing Drawings and Data** group (Sequential UI mode only) creates manufacturing drawings in ISO 10110 format and a Zemax-specific format, as well as exporting data about the surface to cross-check manufacturing setup.

See The Tolerance Tab chapter for full details.

Libraries Tab
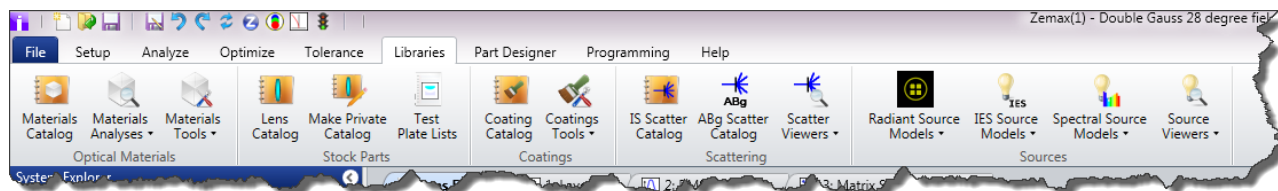
OpticStudio ships with a lot of data on optical materials, stock lenses, thin-film coatings and light sources built in, in the form of catalog data, plus you can add your own data as well. This tab provides access to all the catalog data that ships with OpticStudio and also lets you enter your own data.

The **Optical Materials** group gives access to catalogs of glasses, plastics, birefringent media and more.

The **Stock Lens** group holds all the vendor lens catalogs in Zemax, which can be quickly searched to find the correct lenses for your application.

The **Coatings** group contains data and tools for designing thin-film coatings and applying them to optical surfaces.

The **Scattering** group gives access to Zemax's surface scattering libraries and scatter viewers. The IS Scatter Catalog contains measured data for a range of optical finishes including various matt blacks, ground glass and anisotropic scattering surfaces.

The **Sources** tab gives access to the libraries of Radiant Source (spatial, angular and optionally wavelength) data and IES (angular only) data. Spectral (color) information can be added to data files that do not have it, and there are several source viewer options.

See The Libraries Tab chapter for full details.

Part Designer Tab
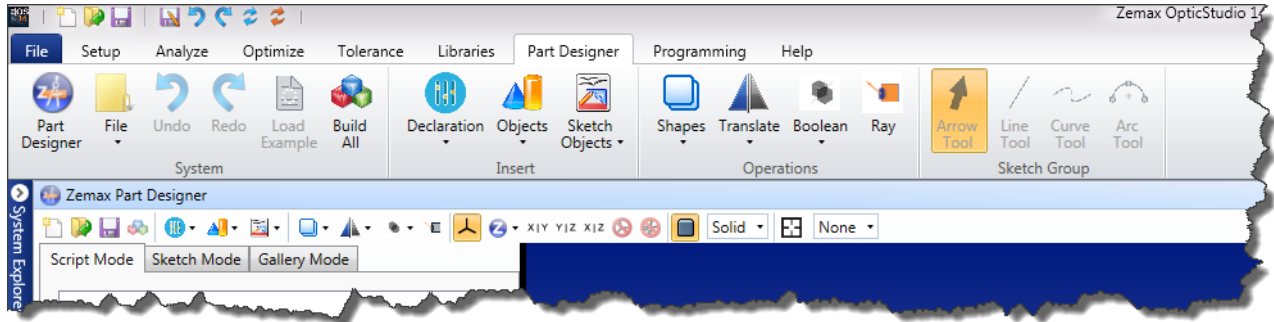
The Part Designer is a sophisticated geometry creation tool capable of making parametric objects which are optimizable within OpticStudio Premium. Part Designer is only available in the Non-Sequential UI mode.

See The Part Designer Tab and Using Part Designer chapters for full details.

Programming Tab
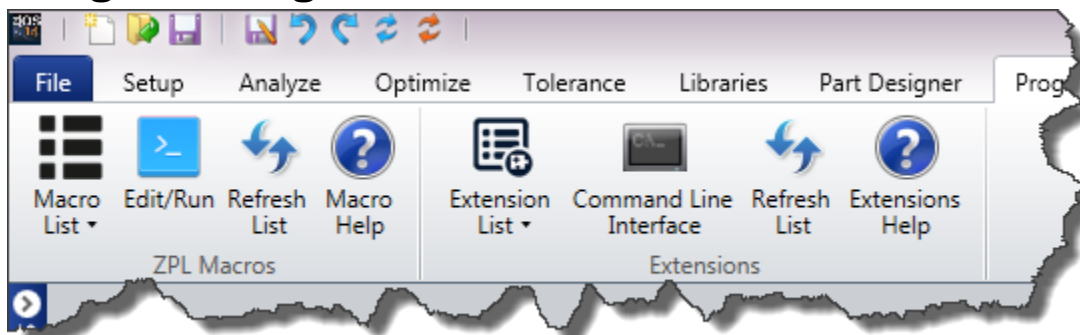
Although OpticStudio has a huge range of features and analysis options, there is always some special feature or requirement. That's why OpticStudio has programming interfaces built right in. The first is Zemax Programming Language (ZPL), which is a very easy to learn scripting language similar to Basic. With ZPL, it is easy to perform special calculations, display data in different ways and automate repetitive keyboard tasks, amongst many others.

'Extensions' are external programs that can take control of Zemax, instruct it to perform analyses and extract data from it. MatLab® and Python are two of the most common programs to work with OpticStudio, plus you can write your own program which can use zclient, our Software Development Kit (SDK) for applications programmers.

See The Programming Tab chapter for full details.

Help Tab
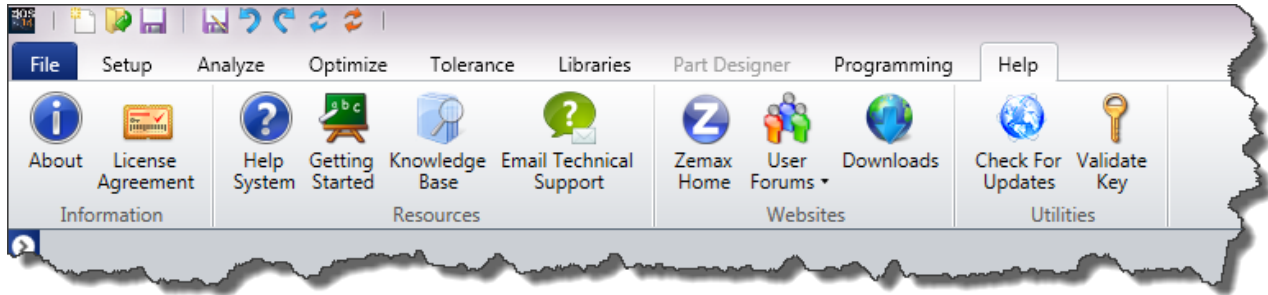
The Help tab provides links to this Help file, plus links to our web-based Knowledge Base, website, and user forums.

See The Help Tab chapter for full details.

Using Spreadsheet Editors

Most data about your optical system is contained in one or more spreadsheets. OpticStudio's spreadsheets are designed to behave like Microsoft Excel® sheets as much as possible, to make them familiar. There are some specific optical system design touches to make you very productive when using OpticStudio. Here is a typical Editor:

	Surf:Type	Comment	Radius	Thickness	Glass	Coating	Semi-Diam	Conic	TCE x 1E-6
0	OBJECT Standard ▾		Infinity	Infinity			Infinity	0.000	0.000
1	Standard ▾		54.153 V	8.747	SK2	AR	29.225	0.000	-
2	Standard ▾		152.522 V	0.500		AR	28.141	0.000	0.000
3	Standard ▾		35.951 V	14.000	SK16	AR	24.296	0.000	-
4	Standard ▾		Infinity	3.777	F5		21.297	0.000	-
5	Standard ▾		22.270 V	14.253		AR	14.919	0.000	0.000
6	STOP Standard ▾		Infinity	12.428			10.229	0.000	0.000
7	Standard ▾		-25.685 V	3.777	F5	AR	13.188	0.000	-
8	Standard ▾		Infinity	10.834	SK16		16.468	0.000	-
9	Standard ▾		-36.980 V	0.500		AR	18.930	0.000	0.000
10	Standard ▾		196.417 V	6.858	SK16	AR	21.311	0.000	-
11	Standard ▾		-67.148 V	57.315 V		AR	21.646	0.000	0.000
12	IMAGE Standard ▾		Infinity	-			24.571	0.000	0.000

All Editors contain three areas:

1. A Toolbar area at the top which provides tools for manipulating the data in the editor and performing tasks specific to that data. For example, the first icon in the Lens Data Editor above is a Tilt/Decenter elements tool which allows any group of elements to be tilted and decentered.
2. A collapsible Properties Inspector that lets you see and set detailed data about each row or item in the editor. This is typically reserved for data that is set once and then not changed frequently. The Properties Inspector can be hidden (collapsed) when not needed.
3. The Spreadsheet region which is where the data you need to work with is kept. Spreadsheet cell data can be entered directly, or can be calculated by Solves, which act similarly to spreadsheet functions in Excel®

Each Editor, its toolbar and Properties is discussed in detail in its own chapter.

For more information see the Using the Editors chapter.

Using Analysis Windows
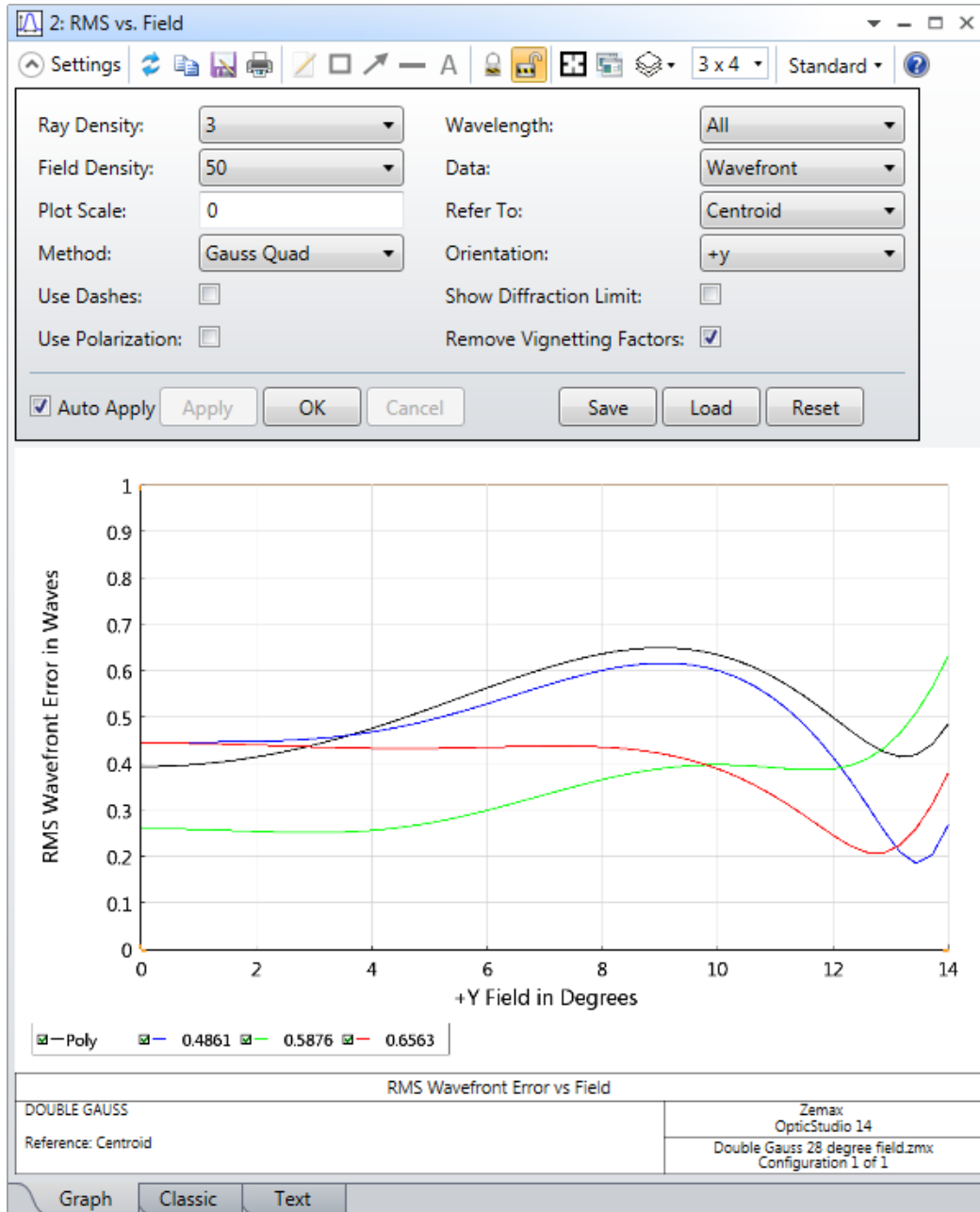
All Analysis windows contain four main areas:

1. A toolbar, which is common across all Analysis windows. This gives access to the Analysis Settings, which control the calculation, and various controls like print, copy, save as etc
2. The Settings dialog, which can be shown or hidden by using the Setting control on the toolbar.
3. The data analysis itself, which can be shown in three ways: as a graphic, as a text listing and (some windows only) in 'Classic' mode which uses the graphics libraries from older versions of Zemax.

4. A context menu that appears when the right-mouse button is clicked:
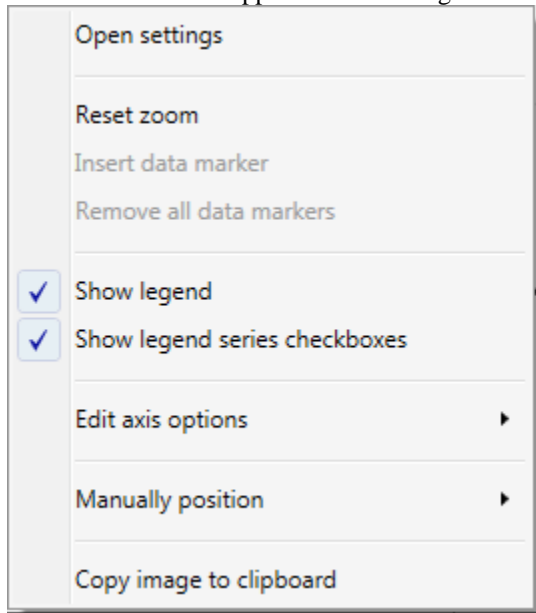
For more information see the [Analyze Tab chapter](#).

Using the Shaded Model
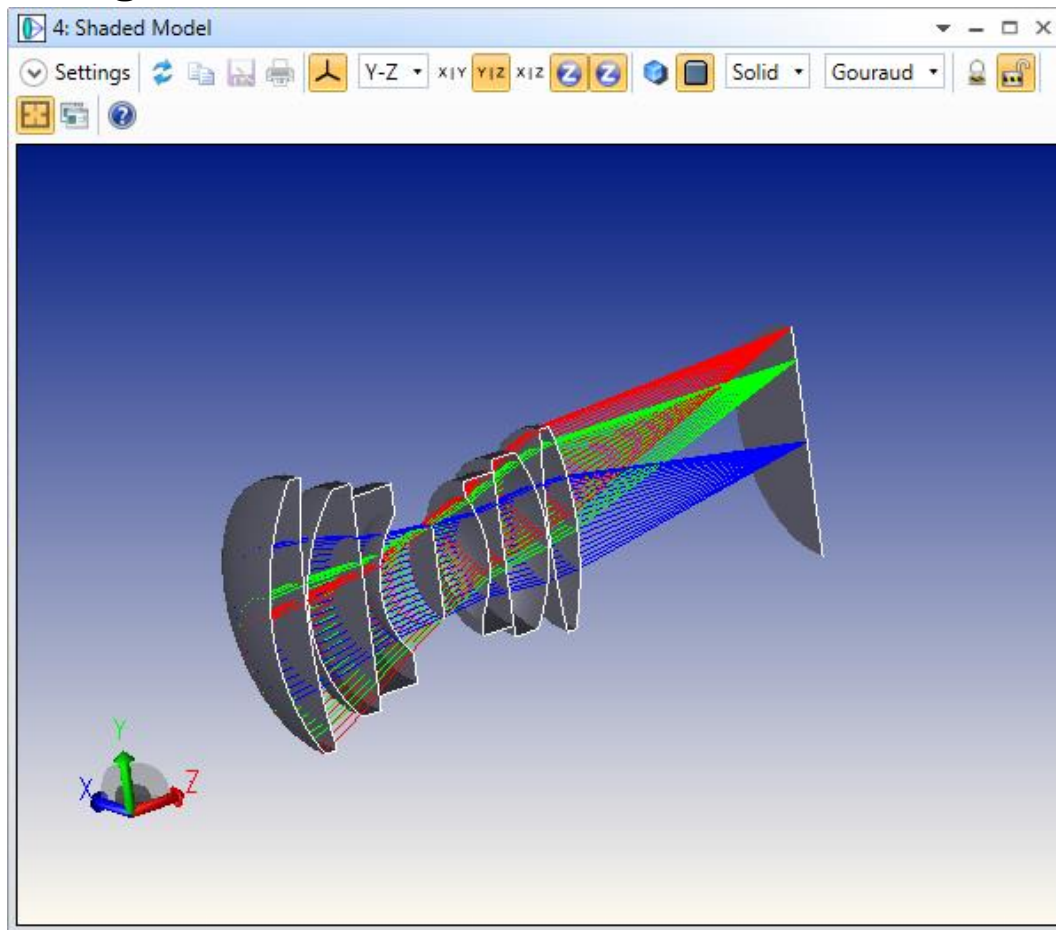
The Shaded Model lets you view your system through an advanced graphics view familiar to anyone who has used modern CAD programs. Mouse control of the view is intuitive, and the toolbar gives access to default views (Isometric, X-Y, Y-Z and Z-Y) and cutting planes which can be used interactively to show the level of detail you need. There are options for wireframe, hidden line or solid modeling views.

See the Shaded Model section of The Analyze Tab chapter for full details.

Arranging Windows

Windows can be free-floating or docked, and docked windows can be tiled in any way you wish. This is best described in a video:

Watch the video: Arranging Windows. (http://zemax.com/products/opticstudio/feature-explorer/user-interface/window-management)

Tutorials

In this section we give several examples of how to use OpticStudio. We strongly recommend that you take the time to work through this section.

If you are viewing this in an HTML dialog, expand the "Tutorials" topic in the Contents tab to view the subsections.

Tutorial 1: A Walk-Through an OpticStudio Design

In this example we open one of the sample files provided with OpticStudio and get familiar with the use of the various controls. Open OpticStudio (click File…New if it is already open) and in the Setup tab choose 'Dock New Windows'
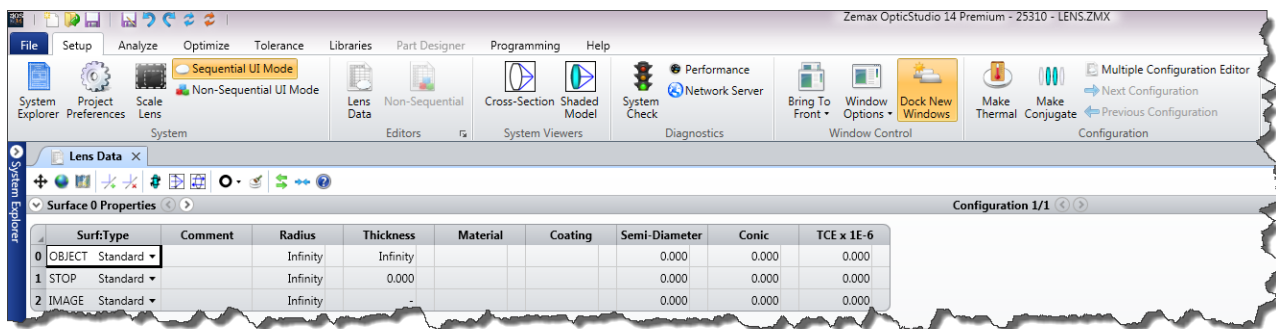
Then click on File…Open and select the file {Current User}\Documents\Zemax\Samples\Sequential\Objectives\Double Gauss 28 degree field.zmx.   This is a classic lens design, and even if you intend to use only the non-sequential mode of OpticStudio you should still follow this example. The user interface is common to both sequential and non-sequential ray-tracing, and users of all versions of OpticStudio will be able to work through this example. The OpticStudio window will look like so:
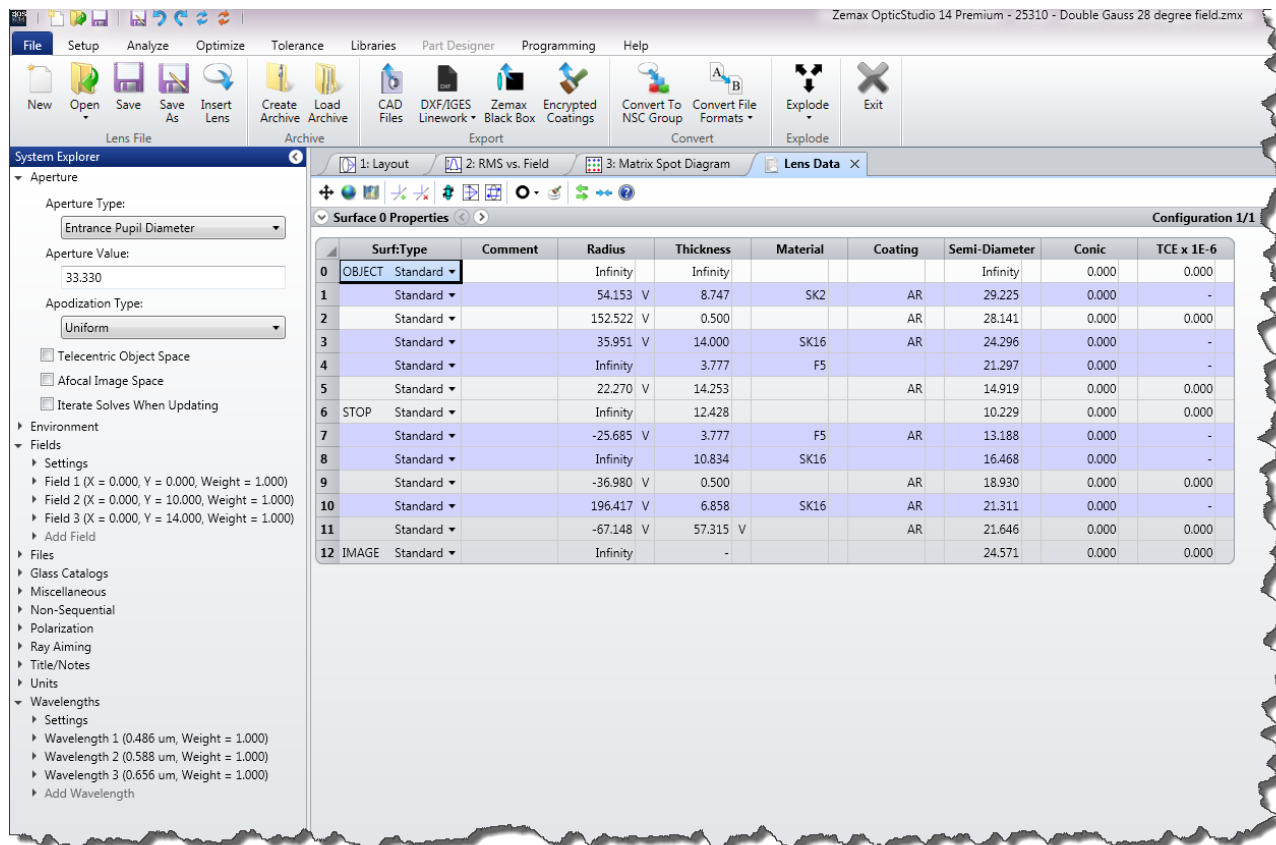
Click on the Layout tab, and right-mouse click on the tab to show the window placement menu:
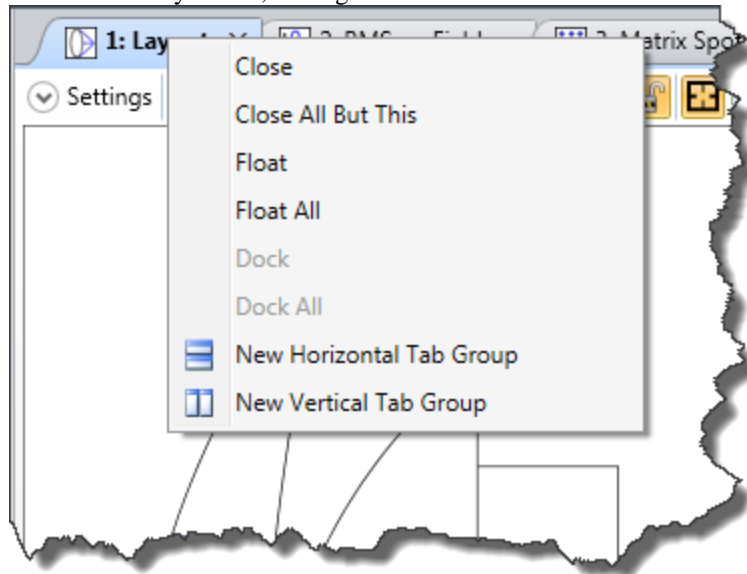
Experiment with floating and docking windows. Then click on the Layout tab with the left-mouse-button and drag the window away from the dock until the Window Manager icon appears. Use this to place the layout below the other three windows:
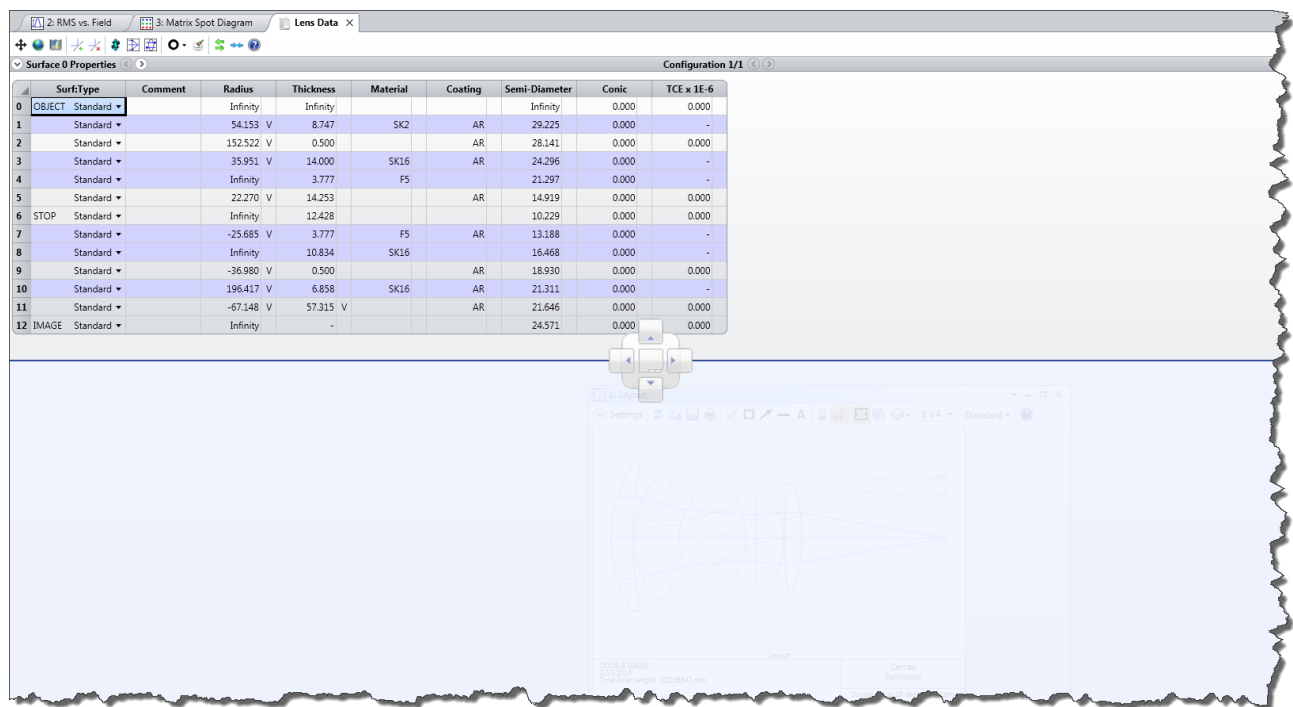
And release the mouse button:
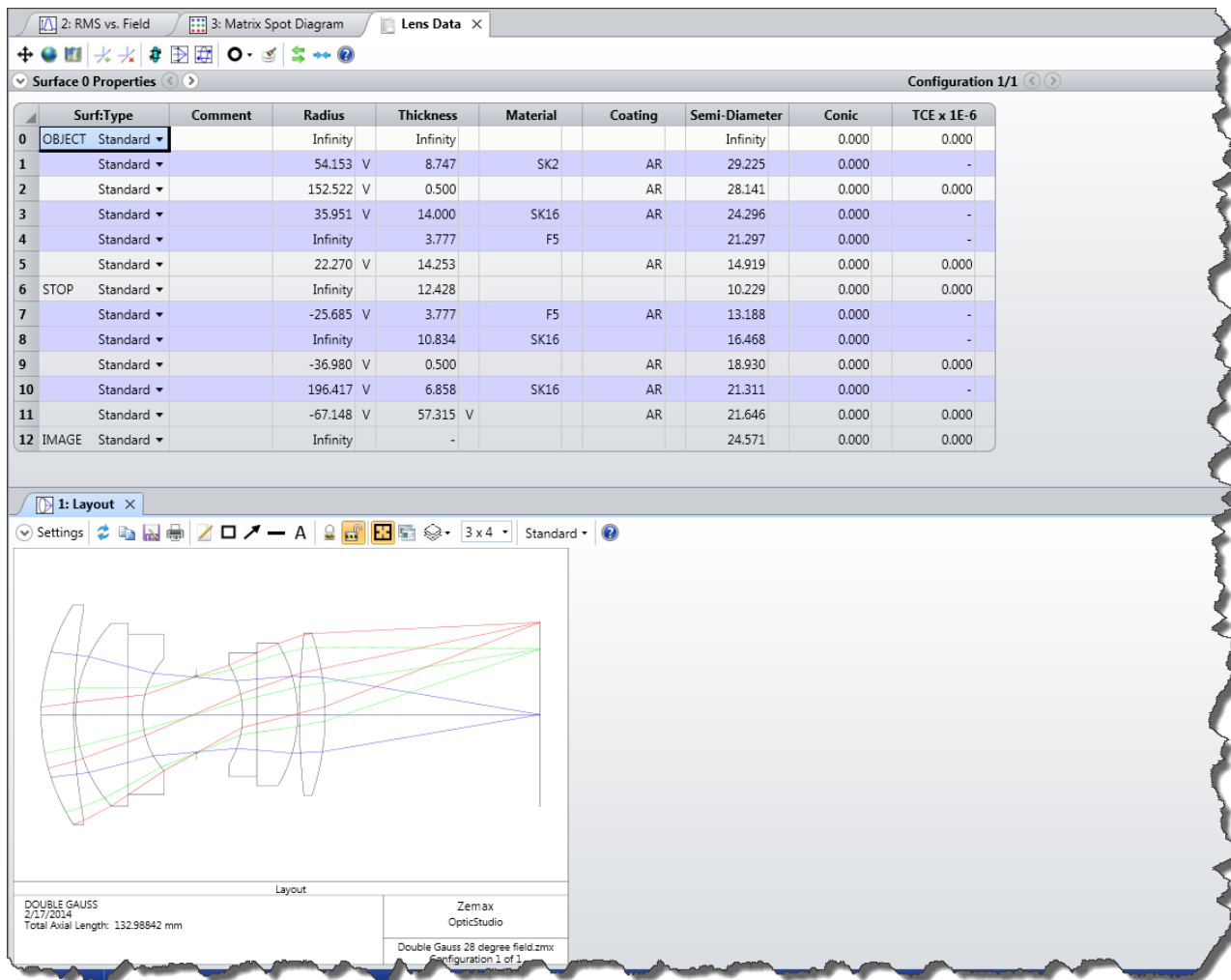
Repeat this process to get the following workspace layout:
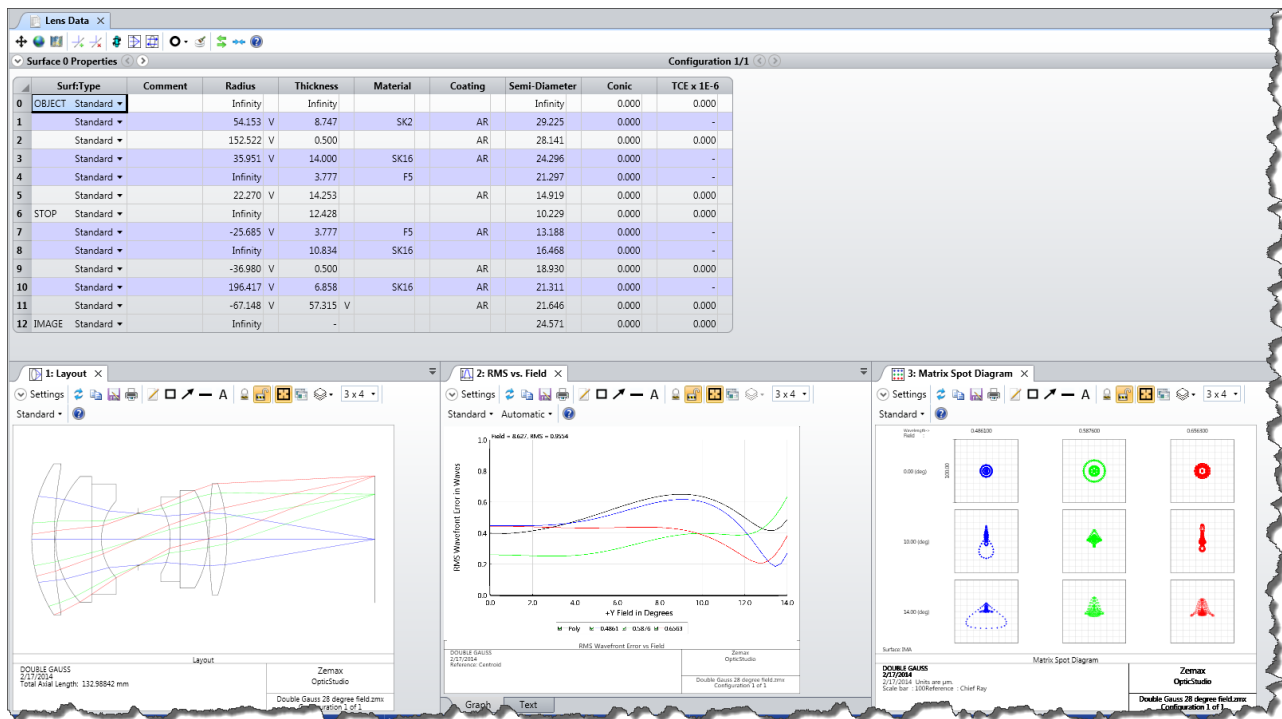
Then make this arrangement:
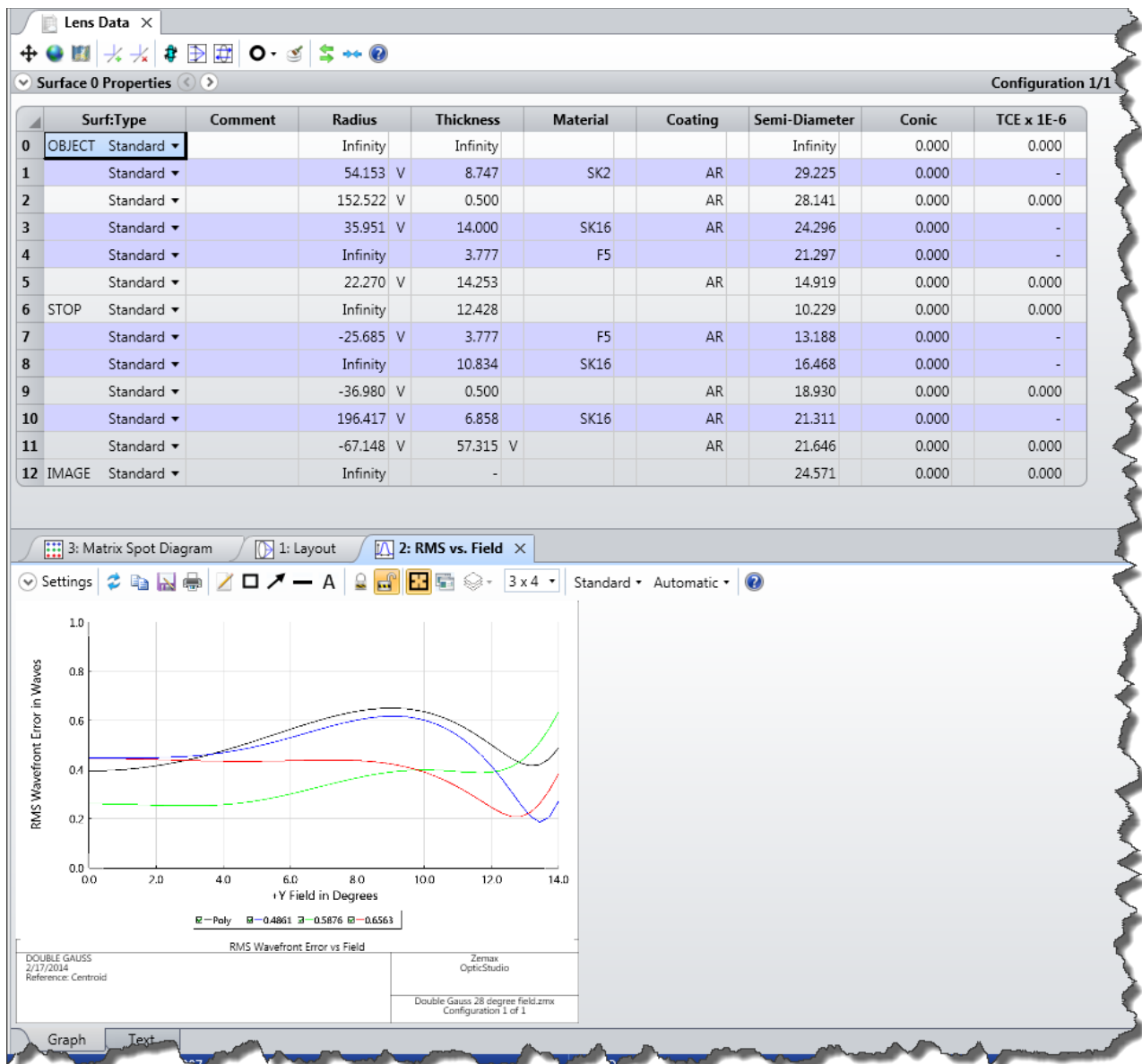
This shows one of the key features of the OpticStudio interface. Any open window many be floating (placed where you put it, and independent of any other window), or arranged in the workspace, or placed in a tab group with any other window etc.

Next, within the Setup tab, click on Performance in the Diagnostics group:
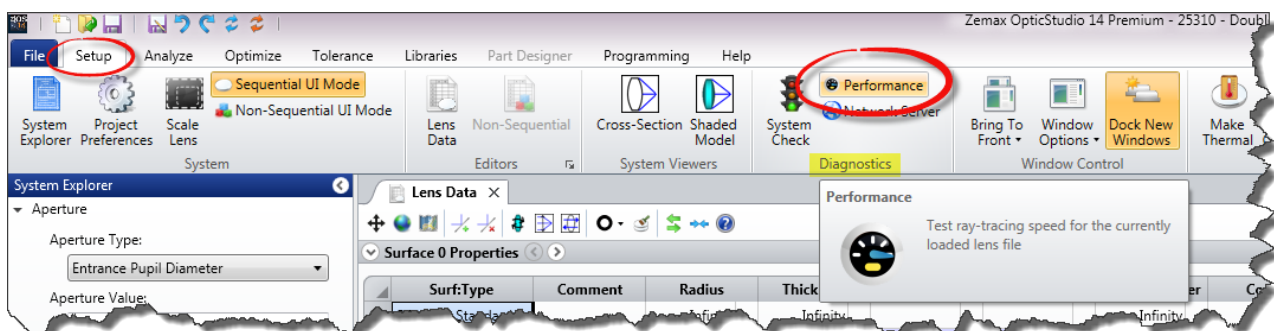
This will give you a simple metric of how fast OpticStudio is on your computer. It also shows one of the best features of OpticStudio: its ability to use multiple CPUs in your computer, if available, to trace millions of rays per second. Calculations are split up and spread over all available CPUs, and the results stitched back together again, without any user interaction.

Tutorial 1.1: The Lens Data Editor

In sequential ray-tracing, light is traced from its source, called the 'Object' surface, to surface 1, then to surface 2, 3, etc. until it lands on the final 'Image' surface. For historical reasons this surface is always called the Image surface, even though the optical system may not form an image of the source. A laser beam expander or eyepiece for example may be afocal: this is covered later.

Surfaces are inserted or deleted in the editor using the Insert or Delete keyboard keys, or via the right-mouse click
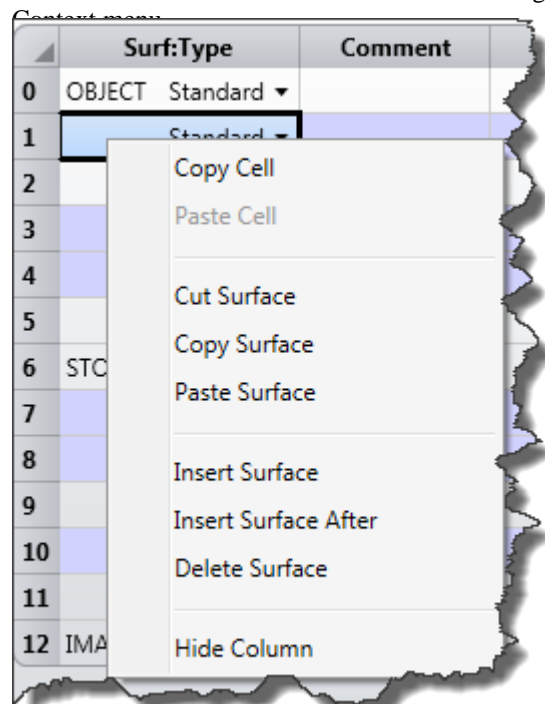
which also allows individual cells or the entire spreadsheet to be copied to the clipboard. Column widths can be varied by placing the cursor in the top row, over the column separator. When the cursor turns to a ↔ symbol, click and hold the left mouse button to resize the column. Columns can be moved by clicking on the column header, holding the mouse down and dragging.

The 'V' next to some parameters means that this parameter is 'variable'. OpticStudio is allowed to change the values in such cells in order to improve the performance. This will be discussed in more detail later.

Surfaces also have a set of properties that are not directly visible in the editor until the Property Inspector is opened. These are generally those properties that are set and then not changed. To see these properties, move the mouse over the Type cell of the chosen surface, and double click.
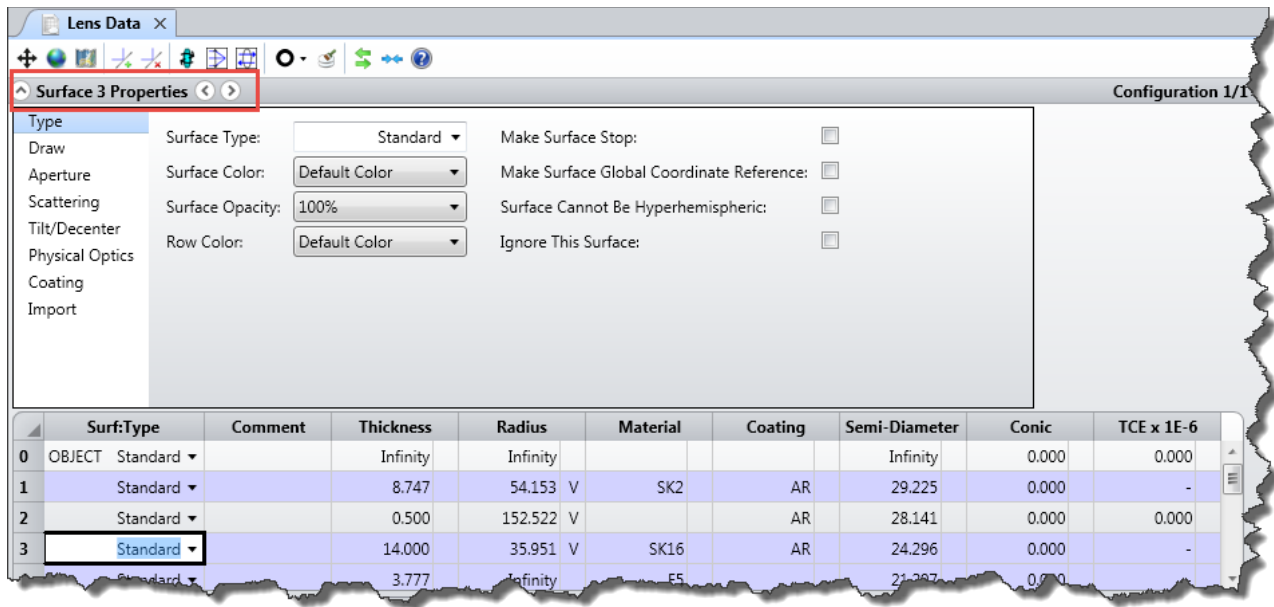
Alternatively, click on the Property Inspector as shown above. A multi-tab dialog appears. From the Surface Type drop-down list you can select the type of the surface, which can be aspheric, diffraction grating, toroidal, etc.

Spend some time exploring each tab. The most commonly used tabs are the Type, Draw, Aperture and the Tilt/Decenter tabs. Press the Help button on each tab to read the on-line Help.

Tutorial 1.2: Analysis Windows

Analysis windows provide either graphical or text-based data computed from the lens as entered in the Editor. Analysis windows never change the lens data: they provide diagnostic information of the various aspects of the lens system's performance.
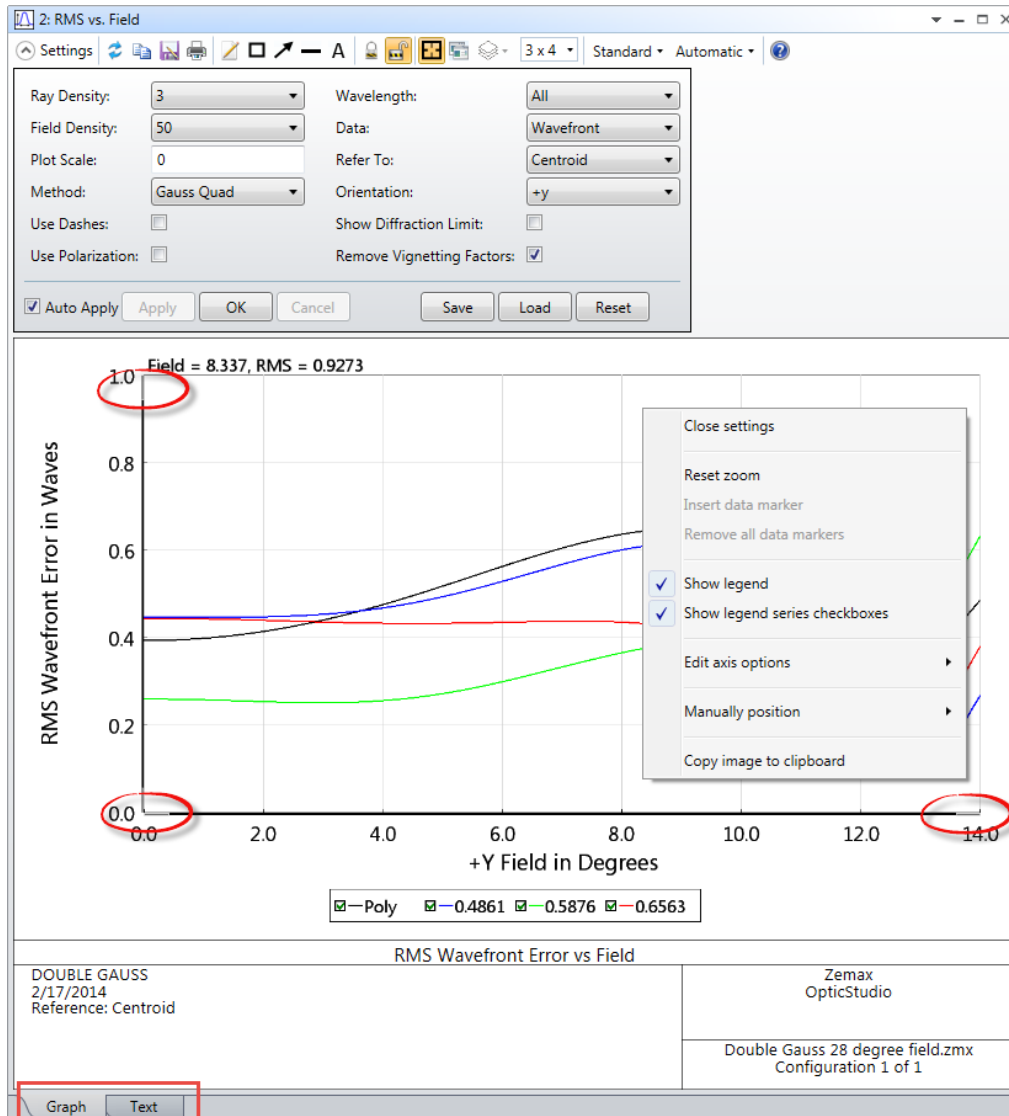
Analysis windows all operate with the same user interface and have the same menu bar:

- Pressing the Settings icon will toggle a dialog box that controls the input parameters passed to the calculation. The Save, Load and Reset buttons allow default settings to be saved, recalled or reset to 'factory' defaults. If you save the settings of any window, those become the defaults for every file that does not have its own settings, so your preferences automatically flow through all your work.

- Pressing the Update menu item (next to the Settings icon), or double-clicking anywhere in the Analysis window with the left mouse button, will make the Analysis window recompute
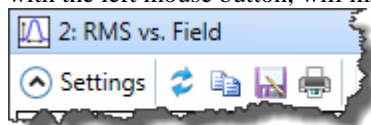
- There are standard Copy, Save As and Print icons

- Right-mouse-clicking will make a context menu show. Analysis windows showing three dimensional data have more options than 2D data plots

- A set of annotation tools let you mark up the analysis feature with text, boxes, arrows, lines etc.:

- Windows can be locked, unlocked, cloned, overlaid and have their drawing resolution changed
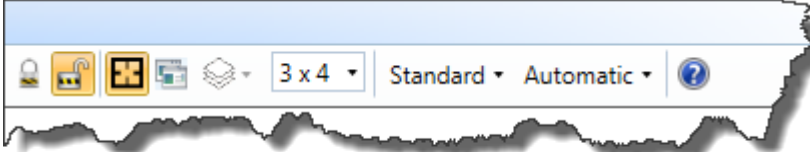
- You can zoom in on a section of interest by clicking the left mouse button, holding it down and dragging it over the region you wish to zoom in on, or by clicking and dragging the grey bars at the ends of the x and y axis, or by choosing 'Edit Axis Options' from the right-mouse menu. A text tab shows the data on which the calculation is based. This can be copied to the clipboard in whole or in part (by highlighting the section desired and right mouse clicking) or saved to a text file.
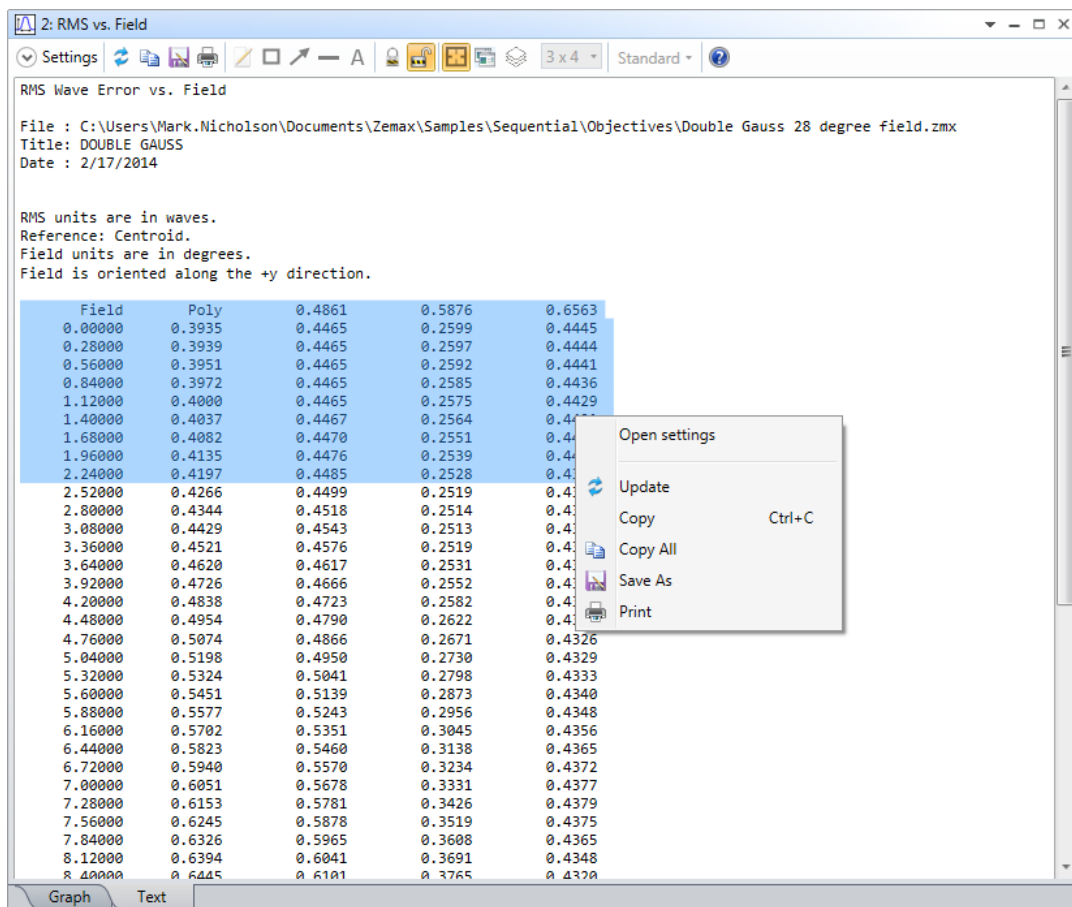
Spend some time playing with the features of the Analysis windows. Remember Analysis windows never change the data in the design, so you cannot do any damage!

Tutorial 1.3: The System Explorer
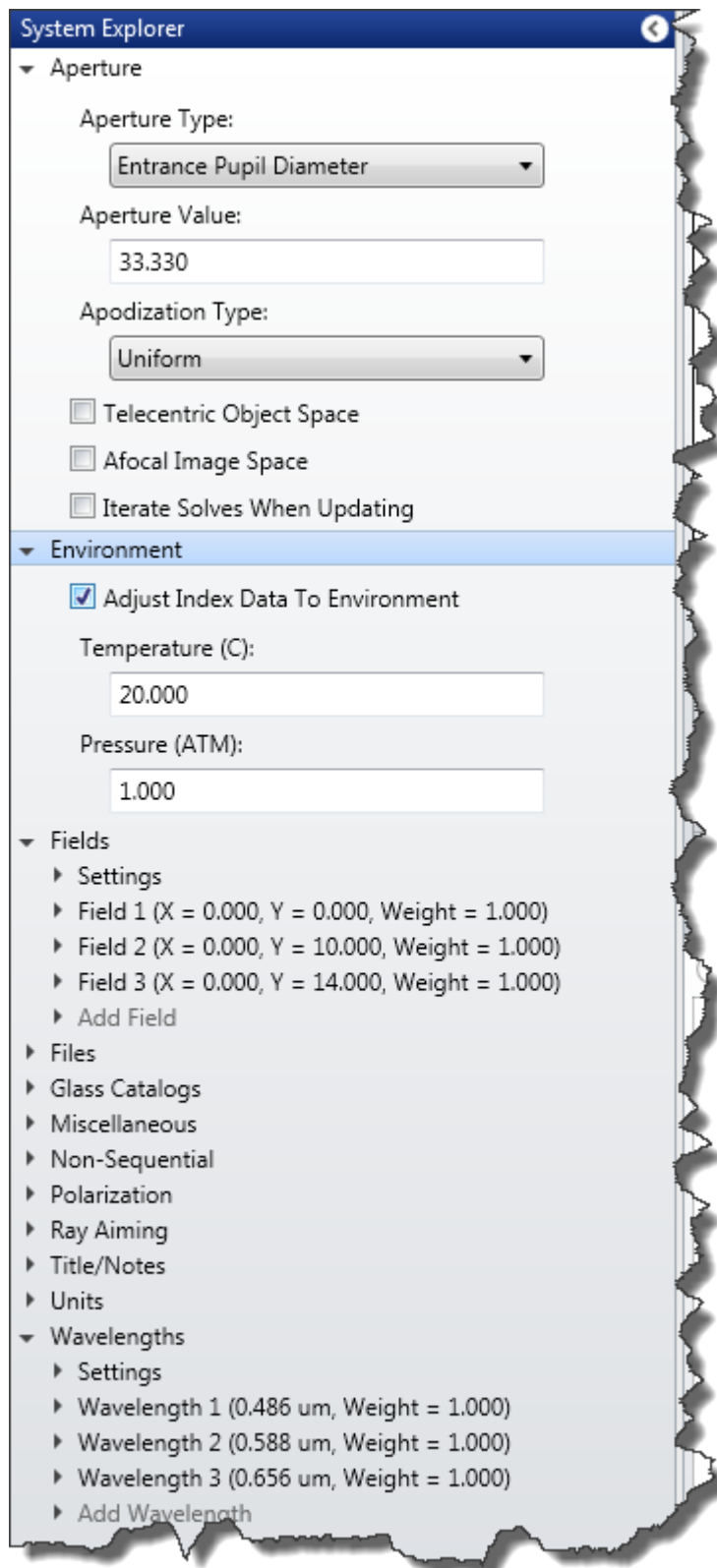
The System Explorer defines the light entering the system as well as many other basic system parameters. Data entered in the System Explorer is usually set once and not altered, or altered only occasionally as the design matures, although full optimization of System Explorer data is supported if needed.
For example, the Aperture of this lens is set to be 'Entrance Pupil Diameter', so the EPD is specified directly, and is set to be 33.33 'Lens Units'. All items having units of length are given in Lens Units except where specifically noted. Click on the Units tab to see what the Lens Units of this lens are (they can be mm, cm, meters or inches).

Entrance Pupil Diameter (EPD) defines the size of the on-axis bundle of light entering the lens system. In the double-Gauss sample file we are using, which is a traditional SLR-type camera lens, OpticStudio traces rays at this height through the lens and computes the size of the aperture stop surface (marked as STOP in the Lens Data Editor), drawn in red opposite. The aperture stop surface is usually a ring diaphragm, so in reality the radial size of this surface defines the EPD, not the other way around.

If you prefer this alternative definition, then choose the Aperture Type in the General dialog box to be 'Float by Stop', and then change the semi-diameter of the STOP surface to say 8 mm. Double-click all the open Analysis windows to make them update to reflect this change, and notice the change in the lens apertures and performance. OpticStudio automatically computes the appropriate size of each surface so that all light passes through each surface.

Another commonly used Aperture Type is 'Object Space NA' which is appropriate when the source is something like an optical fiber that radiates out in a defined numerical aperture. Use 'Object Cone Angle' if the source is defined by a source angle in degrees instead of NA.

The Fields Section

The term "Field" is short for field-of-view and it can be defined in three ways, one of which supports two options:

- The height of the object scene being imaged

- The height of the image being formed, which may be chosen to be either a real or paraxial image height

- The angle subtended by the object scene at the lens
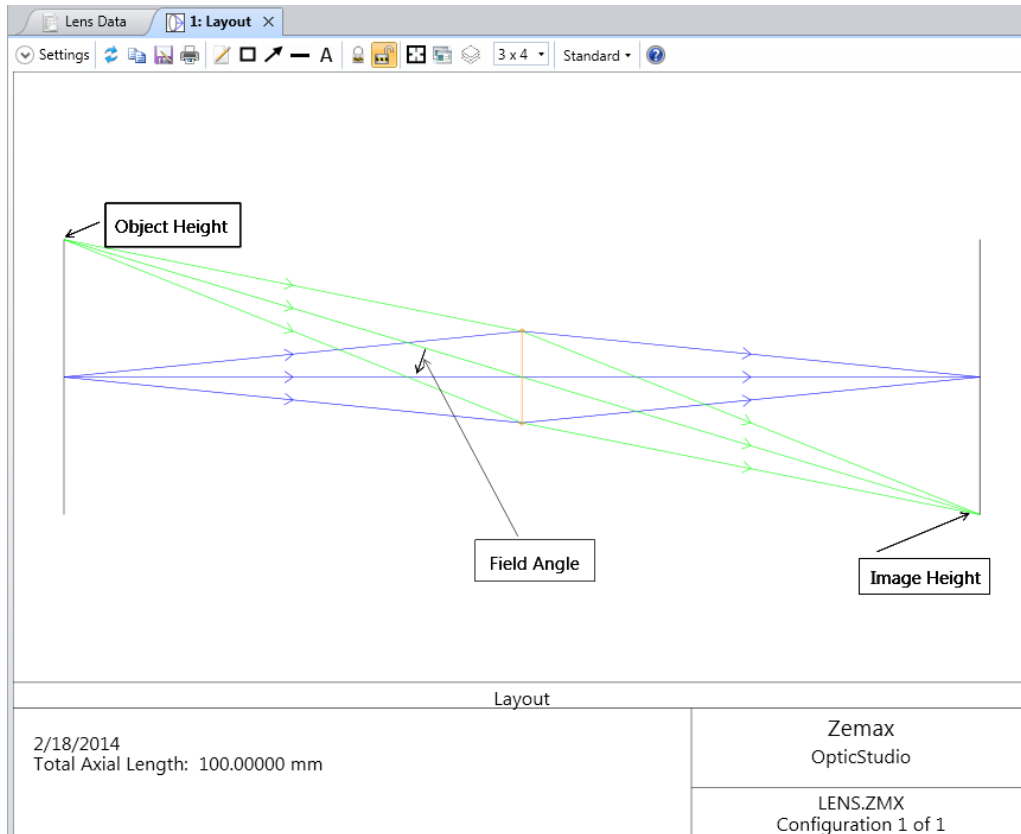
Field data can be entered by expanding the Fields section of the System Explorer, or by double-clicking the Fields data section to show a dialog box:
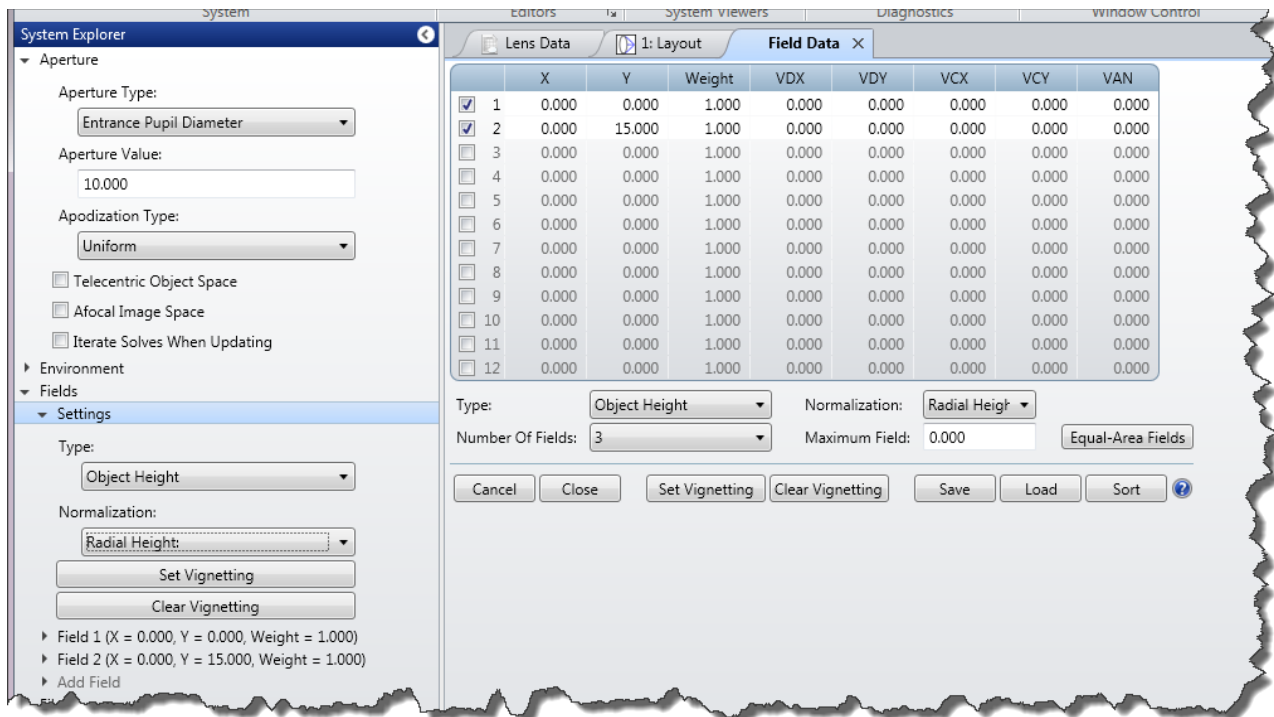
In the double Gauss sample file, we cannot use Object Height as a definition because the object surface is at infinity. Instead, the field angle subtended at the pupil is used, and this is commonly done in camera design. Alternatively, we might use image height when the detector is a fixed size like a CCD chip, and in this case the field of view is defined by what the detector can 'see' when looking out through the lens.

The Wavelengths Section
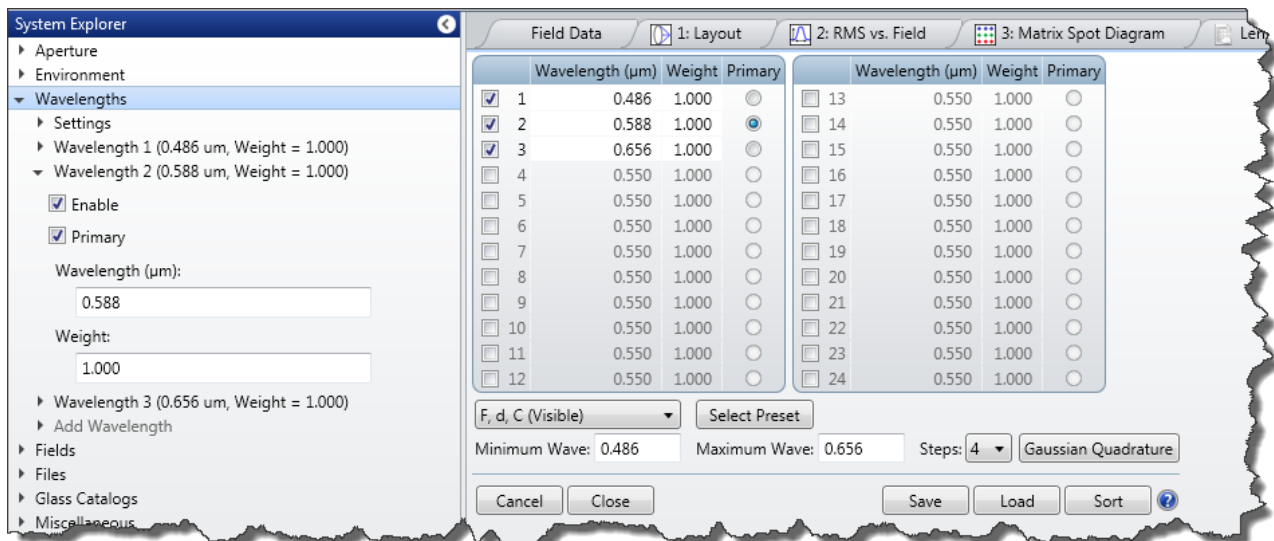
Wavelengths are always entered in microns. Wavelength weights can be used to define relative spectral intensity, or simply to define which wavelengths are most important in a design. The 'primary' wavelength is used as a default wavelength: for example, if asked to compute effective focal length, OpticStudio will compute it at the primary wavelength if no wavelength is specified.

Wavelength data is entered in the System Explorer by expending the Wavelengths data section, or by double-clicking it to give a dialog box.

Tutorial 1.4: The Normalized Coordinate System

Because there are six ways to define system aperture, and four ways to define field of view, it is convenient to work in normalized coordinates. When performing the initial setup of your system you should choose the most appropriate aperture definition, and the most appropriate field definition, and enter the data for both of these. Subsequently, all calculations use normalized units, and you do not have to refer to the specific values entered or definitions used.

Normalized Field Coordinates

Normalized field coordinates Hx and Hy are used throughout OpticStudio, its documentation, and in the wider optical design literature. The normalized field coordinate (0, 1) for example, is always at the top of the field of view in y, whether the field points are defined as angles or heights, and regardless of the magnitude of the field coordinates. Similarly the field coordinate (0,0) is always at the center of the field of view.

For example, suppose 3 field points are defined in the (x, y) directions using object height in lens units of millimeters at (0, 0), (10, 0), and (0, 3). The field point with the maximum radial coordinate is the second field point, and the maximum radial field is therefore 10 mm. The normalized coordinate ($H_x =0$, $H_y = 1$) refers to the location on the object surface (as the field of view is defined in object height) of x= 0, y =10 mm. The normalized coordinate ($H_x = 1$, $H_y = 0$) refers to the object surface location (10, 0).

You can then define any point within the field of view of the lens by its (H_x, H_y) coordinates, as long as $H_x^2 + H_y^2 \leq 1$.

This is referred to as radial field normalization, as the normalized field coordinates represent points on a unit circle. OpticStudio also supports rectangular field normalization, in which the normalized field coordinates represent points on a unit rectangle.
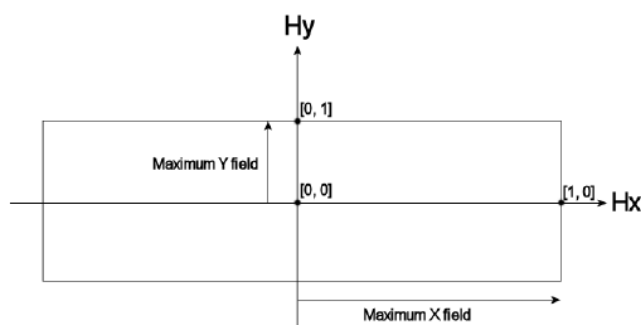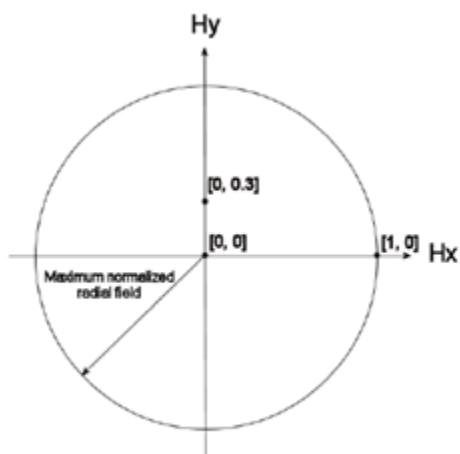
Normalized Pupil Coordinates

Similarly, normalized pupil coordinates are also used throughout OpticStudio, its documentation, and in the wider optical design literature. You define the system aperture using whatever definition is most useful, and thereafter we use the normalized pupil coordinates Px and Py to define any point within a unit circle. Therefore, the point (0,1) represents a point at the top of the bundle of rays entering the system, and (0, 0) is a point at the center of the ray bundle, no matter what the definition of system aperture is or what value it has.

Using the Normalized Coordinates

Re-open the double Gauss 28 degree field sample file in order to undo any changes you may have made in the earlier sections. Open the Field section of the System Explorer and note that the field is defined as angle in degrees, and the maximum field point has a value of 14°. This is a half-angle, and so the full field of view is 28°.

Note: OpticStudio is always clear on the definitions it uses, but these definitions are not universal in the optics industry. Always clarify with your customers what definitions they use for important system specifications to avoid costly errors!

Then expand the Aperture section of the System Explorer and note that the system aperture is defined as Entrance Pupil Diameter, value 33.33. Go to the Units tab to see that the lens units are millimeters, so the EPD is 33.33 mm.

Lastly, expand the Wavelength section and note that the design uses three wavelengths, at 0.4861, 0.5876 and 0.6563 microns respectively. The primary wavelength is set as wavelength number 2, which is 0.5876 microns.

Now click on the Analysis tab, and select Rays and Spots…Ray Trace. This is the most fundamental calculation in OpticStudio: the tracing of a single ray. Right-mouse click on this window to bring up its Settings dialog box:
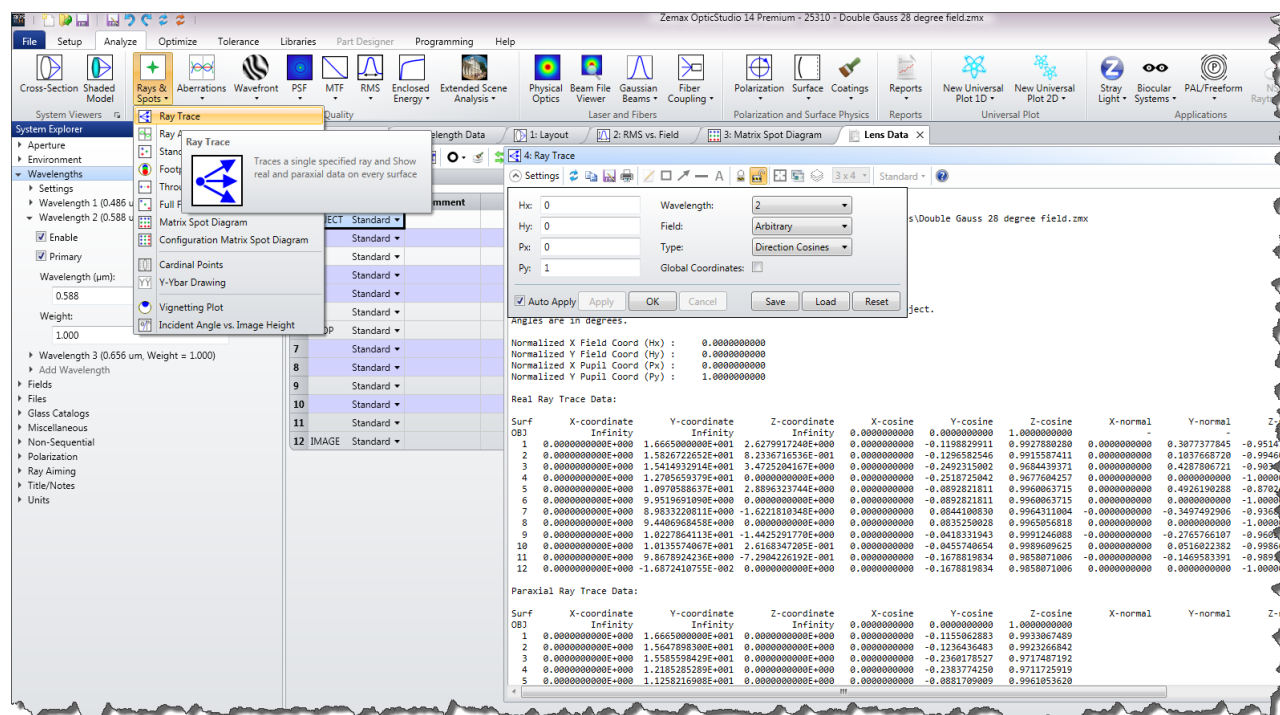
Note how you can define any ray by where it starts on the object surface in normalized (H_x, H_y) coordinates, and where it goes to in the pupil in normalized (P_x, P_y) coordinates. Try tracing some rays, and look at the data provided by this feature. You are given the (x,y,z) position of the ray, the direction cosines of the ray, and the path length of the ray at each surface of the system. This is the fundamental data on which all the calculations in sequential ray tracing are based.

Note also that instead of defining 'arbitrary' field coordinates (H_x, H_y) you can also define the starting coordinates of the ray by the field point number:
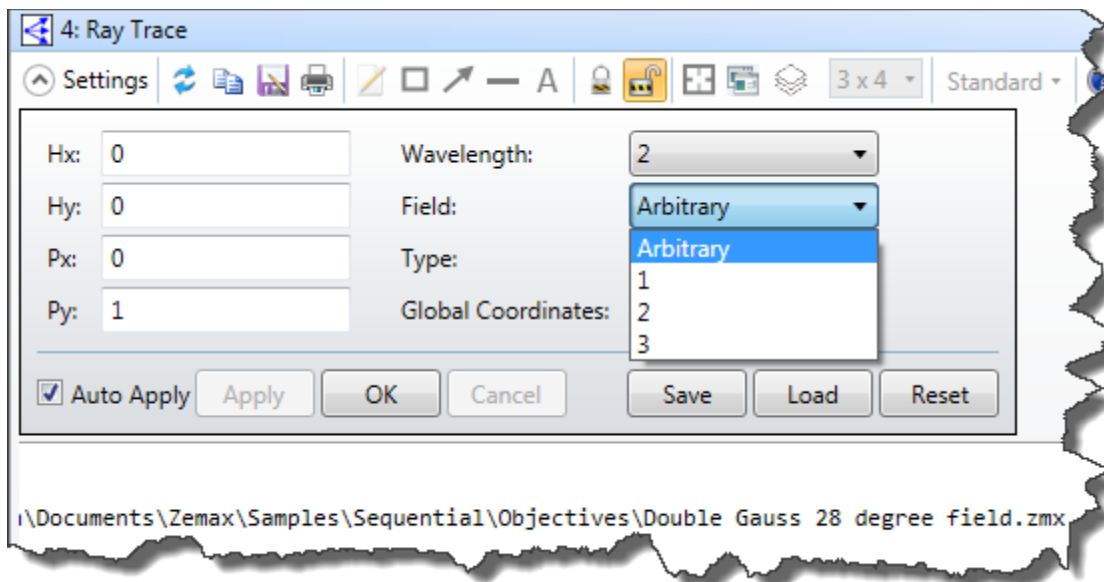
Since field point 3 defines the maximum radial field, it is at (0, 1) in normalized field coordinates. Field point 2 is likewise at (0, .714) and field point 1 is at (0, 0). Depending on the Analysis feature selected, field points can be defined by either their field point number (as entered in the Field dialog box) or by their (H_x, H_y) values.

Tutorial 2: Defining, Positioning and Moving Surfaces

Click on File…Open and select the file {Current User}\Documents\Zemax\Samples\Sequential\Objectives\Double Gauss 28 degree field.zmx. If you already have this file open from Tutorial 1, re-open it to lose any changes you may have made during that tutorial.

In 'sequential' lens design, light always starts at surface 0, the 'object' surface, and is traced to surface 1, then surface 2, then 3 etc. It therefore makes sense to position surfaces relative to each other. Returning to the double Gauss sample file, look at the Lens Data Editor, and click on surface 3. Note that this surface is drawn in red in the Layout plots when you click on it in the editor.

Rays propagate from left to right in the layout, and this direction is the +Z axis. The +Y axis goes from the bottom to the top of the window, and the +X axis goes 'into' the screen. If you position your right hand, such that your middle finger is touching surface 1 in the layout window and points into the screen, your index finger is pointing towards the right hand of the screen, and your thumb is pointing upwards, you have the classic 'right hand coordinate system' used throughout OpticStudio and in most of the optical design literature. The Coordinate Axes are always drawn in the Shaded Model plot, below

Also, observe that as you move your mouse over the layout window, the coordinates of the mouse pointer are shown in the title bar of the window:
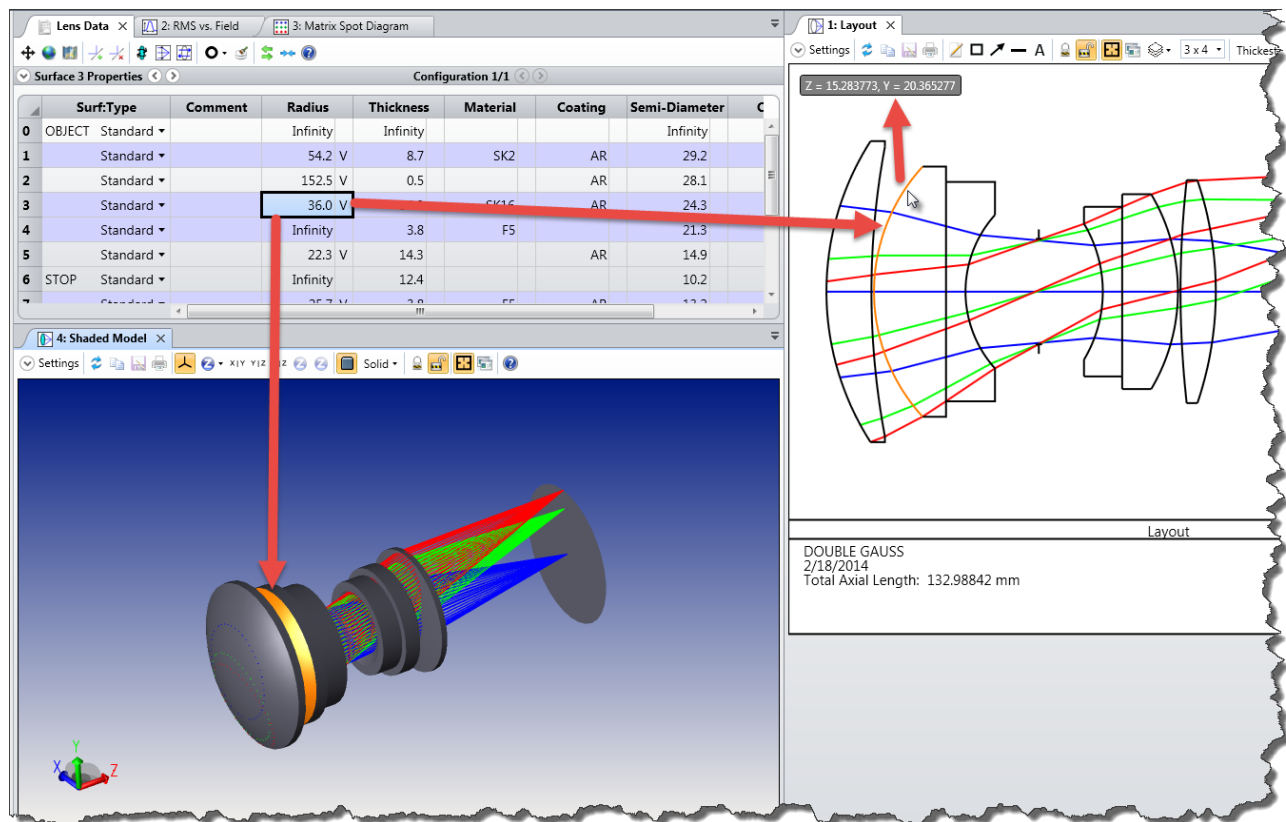
If you move the mouse across the Layout window, you will see the Z coordinate change, and as you move the mouse up or down the window, the Y value will change.

The Lens Data Editor shows the following data for surface 1 (if you are using the Demonstration version of OpticStudio, fewer decimal places are shown:

```
Radius of curvature:        54.153
Thickness:                  8.747
Material:                   SK2
Semi-diameter:              29.225
Conic:                      0
```

Remember that any lengths where the units are not explicitly given are in "Lens Units", which are in mm for this file (see System Explorer 'Units' section).

The radius of curvature of surface 1 is 54.153 mm. This is a positive number because the center of curvature lies in positive Z. By contrast, surface 7 has a radius of curvature of -25.685 mm because its center of curvature lies in –Z.

Surface 1's thickness of 8.747 mm means that surface 2 is located 8.747 in positive Z relative to surface 1. "Thickness" is therefore the distance along Z of two surfaces. We refer to this as 'thickness' in the editor rather than 'z-distance' because if you were to hold the lens formed by surfaces 1 and 2 in your hand, you would naturally describe the lens as having a center thickness.

Surface 1's Material type is set to SK2, which means that the space between surface 1 and surface 2 is filled by a particular glass called SK2. Click on the SK2 glass in the editor, and then click on the Libraries tab and then the Materials Catalog icon:
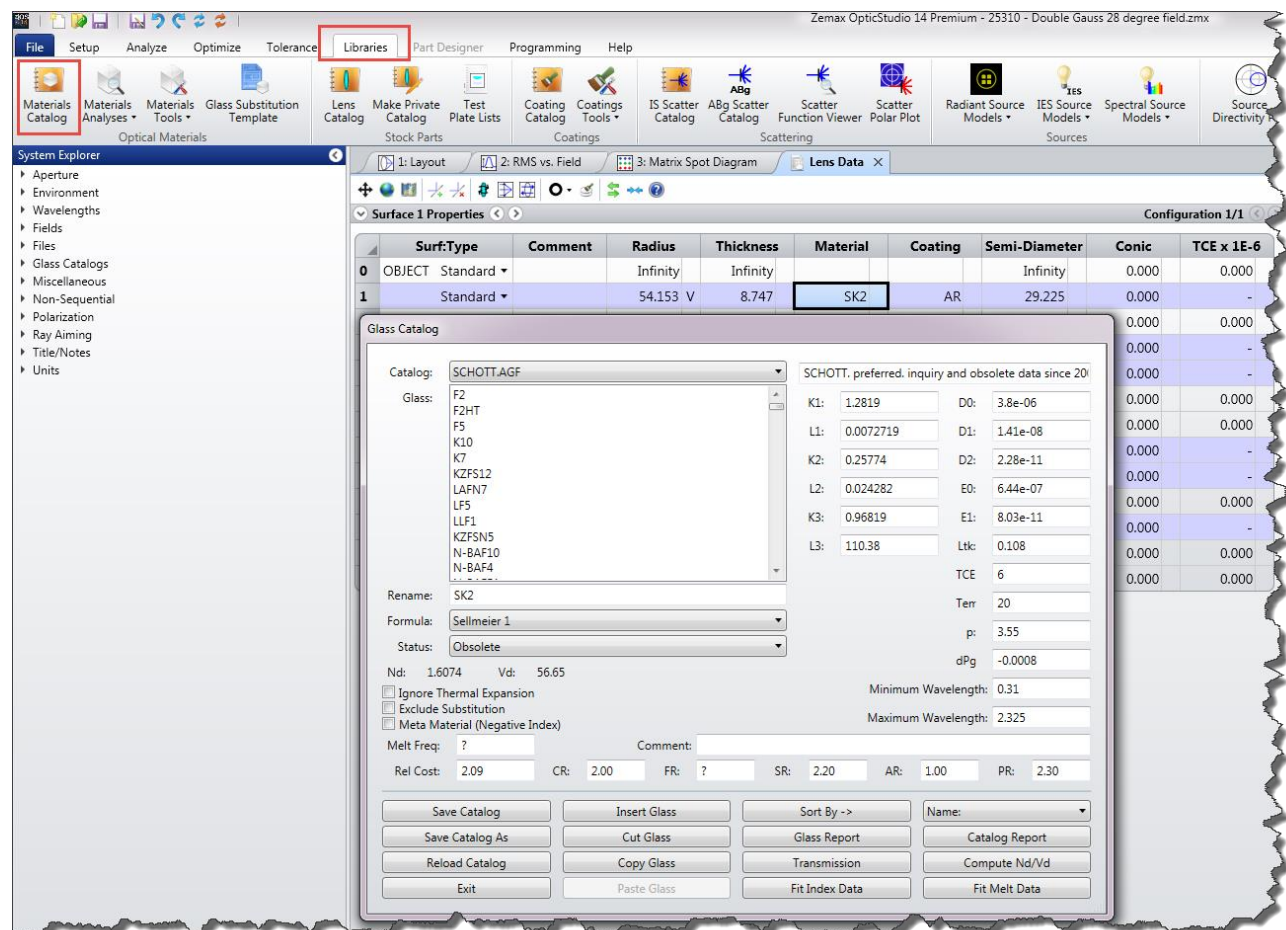
This shows that SK2 is a material in the Schott catalog, and it gives all the data OpticStudio knows about the glass.

Note: A full discussion of this dialog is outside of the scope of this Tutorial. See the "Using Glass Catalogs" section of the Technical Reference for full details.

Next, see that surface 2 has no entry for its Glass column, and is instead blank. That means that surface 2 is made of 'Air' at Standard Temperature and Pressure. Both temperature and pressure can be changed, both at a system level and

on a surface-by-surface basis. This has important, but subtle, effects. First, the index of refraction of glass depends upon both temperature and wavelength; relative indices which are measured with respect to air also change with pressure. Second, glass expands and contracts with temperature, which can change the radius, thickness, or other dimensions of a lens. Third, the distances between lenses change due to the expansion and contraction of the mounting material.

The thermal analysis features of OpticStudio can account for all these effects. OpticStudio can be used to analyze and optimize a design for any specific temperature or for a range of temperatures. This is outside of the scope of this Guide, however, and we will assume that the whole lens is at 20° C, 1 Atm pressure.

Note: See the "Thermal Analysis" section of the Technical Reference for full details of the comprehensive temperature and pressure modelling capabilities of OpticStudio.

The "Semi-Diameter" column shows the radial height of the surface (it is called Semi-Diameter to avoid confusion with radius of curvature). This can be computed in two ways: automatically by OpticStudio (the default), or directly entered by the user.

The automatic calculation sets the semi-diameter to ensure that the edge rays always pass through the lens. This means that the lenses are 'just big enough' to pass the full aperture of rays from across the field of view. Usually, lenses are made a little larger than this, so there is some unused glass that can be used to hold the lens in its mount without blocking the beam. You can specify the additional amount easily by adding a semi-diameter margin in System Explorer's Miscellaneous group.
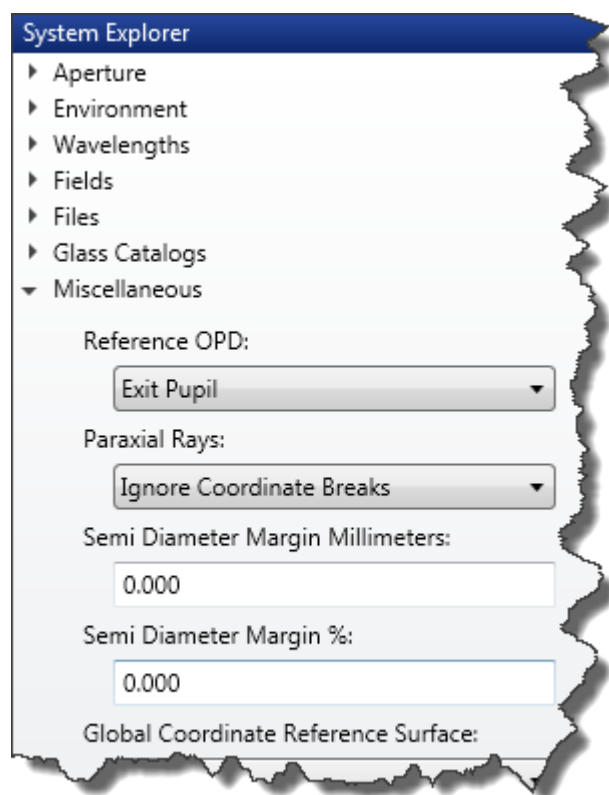
Tutorial 2.1: Working in Three Dimensions

The double Gauss example lens is an axially symmetric lens, and so each surface is simply positioned a distance in Z away from the previous surface. But what about systems in which optical components are tilted or decentered with respect to each other?

Let's imagine that the second group of elements (the doublet and singlet after the stop) is tilted and decentered with respect to the first. Click on surface 7 in the Lens Data Editor, then hold the left-mouse button down while dragging with

the mouse to highlight surfaces 7 to 11. Alternatively, click on surface 7, and press the Shift key while also pressing the down cursor key to highlight surfaces 7-11.

			Thickness	Material	Coating	Semi-Diameter	Conic	TCE x 1E-6
0			Infinity			Infinity	0.000	0.000
1		V	8.747	SK2	AR	29.225	0.000	-
2	Standard ▼	152.522 V	0.500		AR	28.141	0.000	0.000
3	Standard ▼	35.951 V	14.000	SK16	AR	24.296	0.000	-
4	Standard ▼	Infinity	3.777	F5		21.297	0.000	-
5	Standard ▼	22.270 V	14.253		AR	14.919	0.000	0.000
6 STOP	Standard ▼	Infinity	12.428			10.229	0.000	0.000
7	Standard ▼	-25.685 V	3.777	F5	AR	13.188	0.000	-
8	Standard ▼	Infinity	10.834	SK16		16.468	0.000	-
9	Standard ▼	-36.980 V	0.500		AR	18.930	0.000	0.000
10	Standard ▼	196.417 V	6.858	SK16	AR	21.311	0.000	-
11	Standard ▼	-67.148 V	57.315 V		AR	21.646	0.000	0.000
12 IMAGE	Standard ▼	Infinity	-			24.571	0.000	0.000

Tilt/Decenter Elements

Add tilt and decenter to the specified range of surfaces

In the Lens Data Editor's Toolbar, click the Tilt/Decenter Elements icon and configure it as shown:

Tilt/Decenter Element

First Surface: 7   Last Surface: 11

Decenter X: 2   Tilt X: 5

Decenter Y: 0   Tilt Y: 0

Order: Decenter then tilt   Tilt Z: 0

Coordinate Break Color: Default Color

Coordinate Break Comment: Tilt/Decenter the second group

☐ Hide Trailing Dummy Surface

OK   Cancel
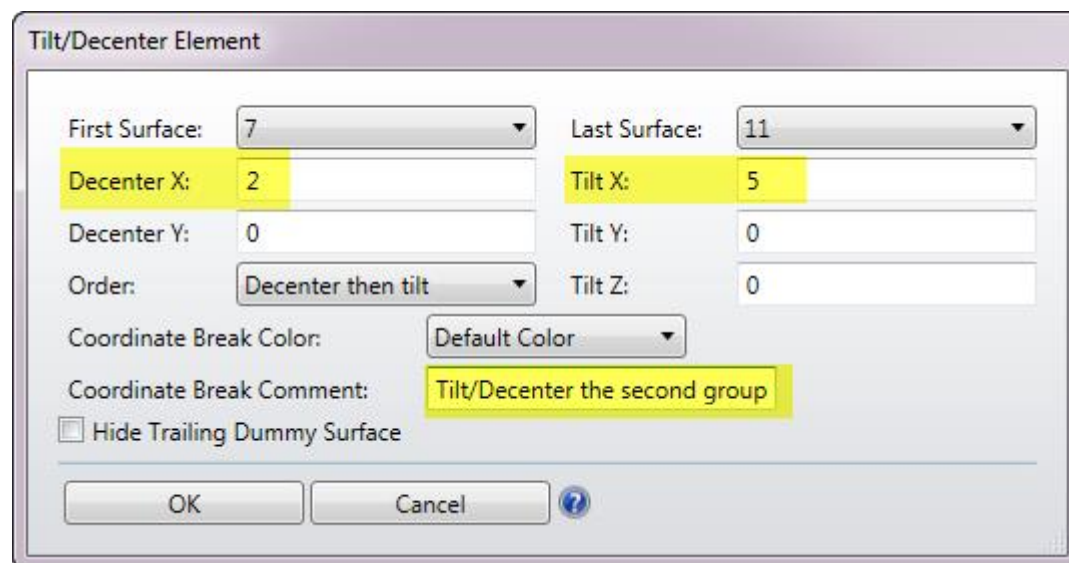
After you press OK, click on System…Update All to update all the open windows. Note that the Layout plot displays a warning 'Cannot perform 2D layout on non-axial system'. Close this window, and click on the Analyze Tab, System

Viewers group and choose the Shaded Model. Configure it like so:
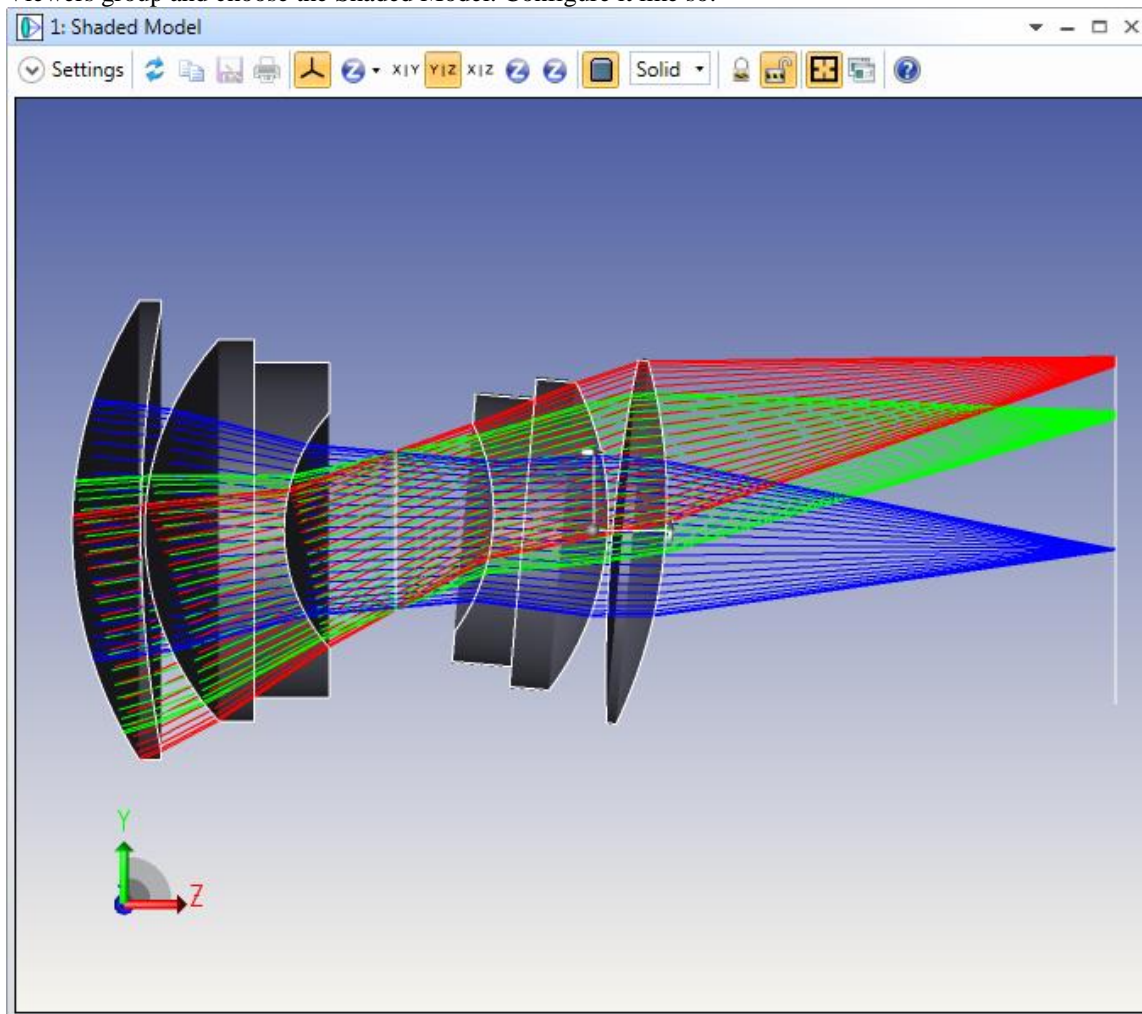
You can see that the second group of elements has been shifted in the y-direction (upwards in the Shaded Model) and rotated around x (the x-axis points into the screen). Click anywhere inside the 3D Layout window so that this window is active (the title bar will appear brighter than the other OpticStudio Analysis windows). The layout can be rotated by using the mouse or by the Rotation X/Y/Z controls in the Settings dialog box for this window.

Return to the Lens Data Editor, and see that OpticStudio has now entered two new surfaces into the design and that the editor now has additional columns.

	Surf:Type	Comment	Radius	Thickness	Material	Coating	Semi-Diameter	Conic	TCE x 1E-6	Decenter X	Decenter Y	Tilt About X	Tilt About Y	Tilt About Z	Order
0	OBJECT Standard ▾		Infinity	Infinity			Infinity	0.000	0.000						
1	Standard ▾		54.153 V	8.747	SK2	AR	29.225	0.000	-						
2	Standard ▾		152.522 V	0.500		AR	28.141	0.000	0.000						
3	Standard ▾		35.951 V	14.000	SK16	AR	24.296	0.000	-						
4	Standard ▾		Infinity	3.777	F5		21.297	0.000	-						
5	Standard ▾		22.270 V	14.253		AR	14.919	0.000	0.000						
6	STOP Standard ▾		Infinity	12.428			10.229	0.000	0.000						
7	Coordinate Break ▾	Tilt/Center the Second Group of		0.000		-	0.000		-	2.000	0.000	5.000	0.000	0.000	0
8	Standard ▾		-25.685 V	3.777	F5	AR	13.422	0.000	-						
9	Standard ▾		Infinity	10.834	SK16		17.280	0.000	-						
10	Standard ▾		-36.980 V	0.500		AR	19.839	0.000	0.000						
11	Standard ▾		196.417 V	6.858	SK16	AR	23.126	0.000	-						
12	Standard ▾		-67.148 V	-21.969 T		AR	23.380	0.000	0.000						
13	Coordinate Break ▾	Tilt/Center the Second Group of:return		21.969 P	-		0.000		-	-2.000 P	0.000 P	-5.000 P	0.000 P	0.000 P	1
14	Standard ▾	Dummy	Infinity	57.315 V	P		21.633	0.000	0.000						
15	IMAGE Standard ▾		Infinity	-			22.126	0.000	0.000						

These are called Coordinate Break (CB) surfaces. Coordinate Break surfaces have no optical effect; they just define a new coordinate system relative to the previous surface's coordinate system. Click on the first CB surface, and scroll to the right in the Editor. See that the surface has a decentration in x, then y, and a tilt in x, y, and z. If you filled in the dialog as shown above, you should see a decentration in y of 2 mm, a tilt about x of 5 degrees, and have an order flag of 0. This means that the CB will execute 'left to right', meaning that the decentration in x is done first, then the decentration in y, then the tilt in x, etc.

Now look at the second CB. It has a decentration in y of -2 mm and a tilt in x of -5 degrees. This 'undoes' the tilt/decentration of the first CB. Also its order flag is not zero, so it executes 'right to left'. This means that the second CB undoes the first CB, and restores the coordinate system to its original frame of reference.

Note also the letter P next to the decenters and tilts of the second CB. This indicates that a 'Pick-Up Solve' has been placed on this parameter. Click on the P symbol of the Tilt-X parameter of the second CB, and the pickup dialog will open:
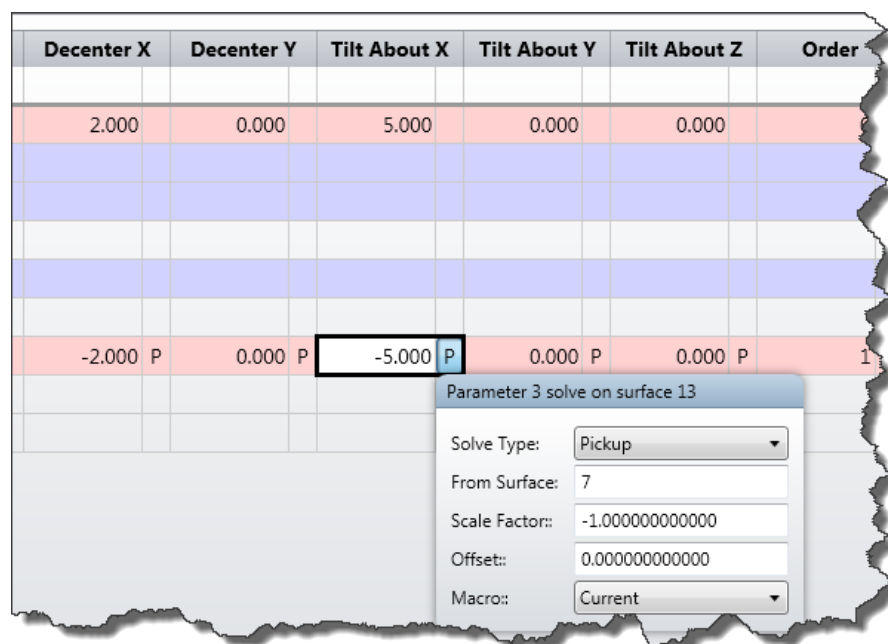
This locks the value of parameter 3 on surface 13 to be whatever the value of parameter 2, surface 7 is, except the sign is the opposite. The pick-up solve is one of the Editor's most useful features, as it allows one part of an optical system to be locked to another.

Note: If you visit the OpticStudio User's Knowledge Base at www.zemax.com/kb, look at the category Sequential Ray Tracing/3D Geometries for many helpful articles about the use of Coordinate Break surfaces.

Tutorial 3: A Multi-Configuration Design

OpticStudio can also work with multiple configurations, or versions, of a design. This is typically used to model zoom lenses (where the spacings between lenses changes), systems where the temperature changes, and systems where the angle of a scanning mirror changes, amongst many others. We will use this now to model a system in which the spot produced by a catalog lens is scanned by a mirror over the image surface.

OpticStudio includes the lens catalogs of all the major vendors so you can easily find catalog lenses. Click File…New to clear OpticStudio, and click on Libraries…Lens Catalog:
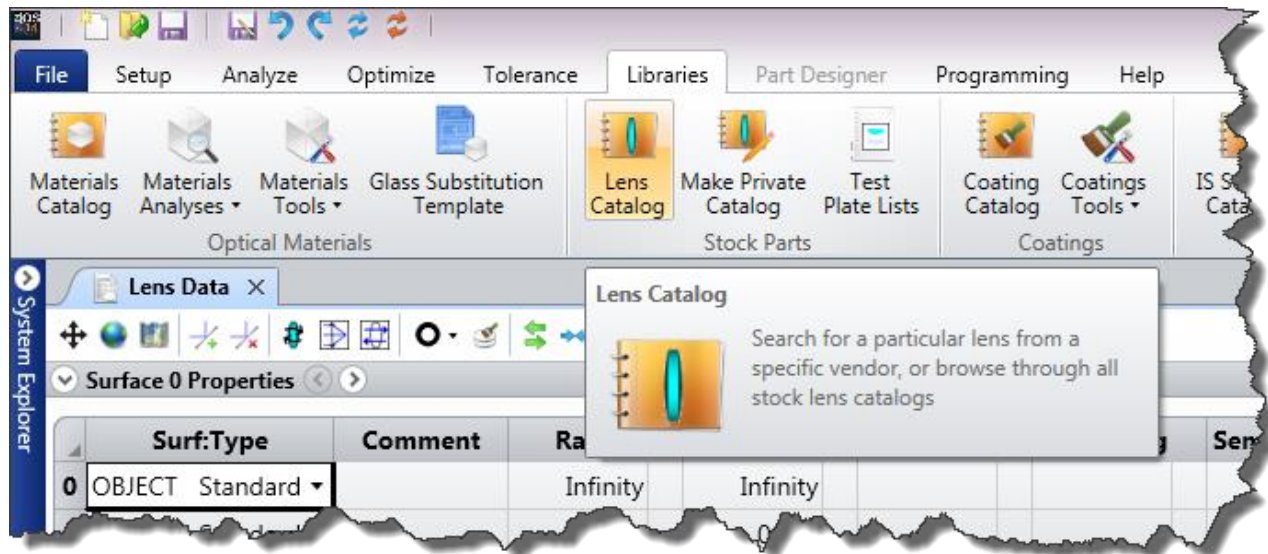
and configure the dialog box like so:
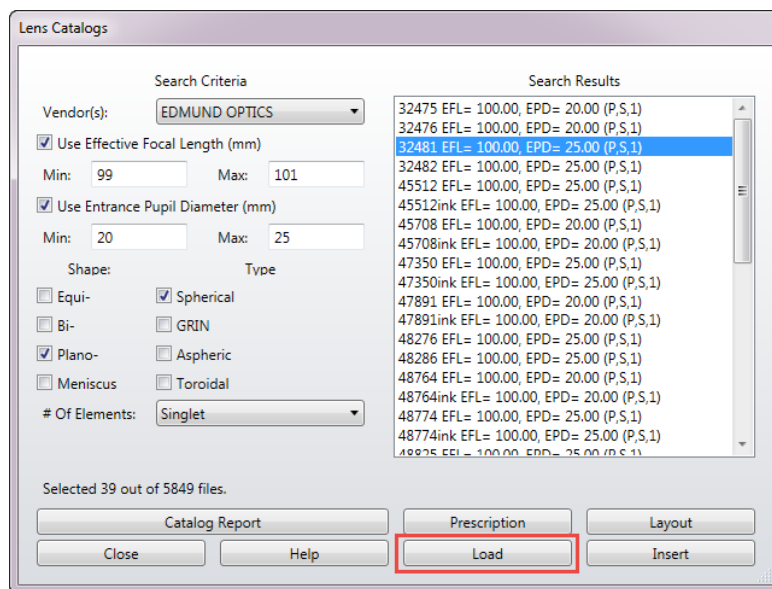
This searches for plano-convex singlet lenses with focal lengths between 99 and 101 mm, and Entrance Pupil Diameters between 20 and 25 mm. Note that there are 27 lenses that meet these criteria out of the 4060 files included in the Edmund Optics catalog. Select the lens 32481, and press the Load button. OpticStudio then loads this stock lens:
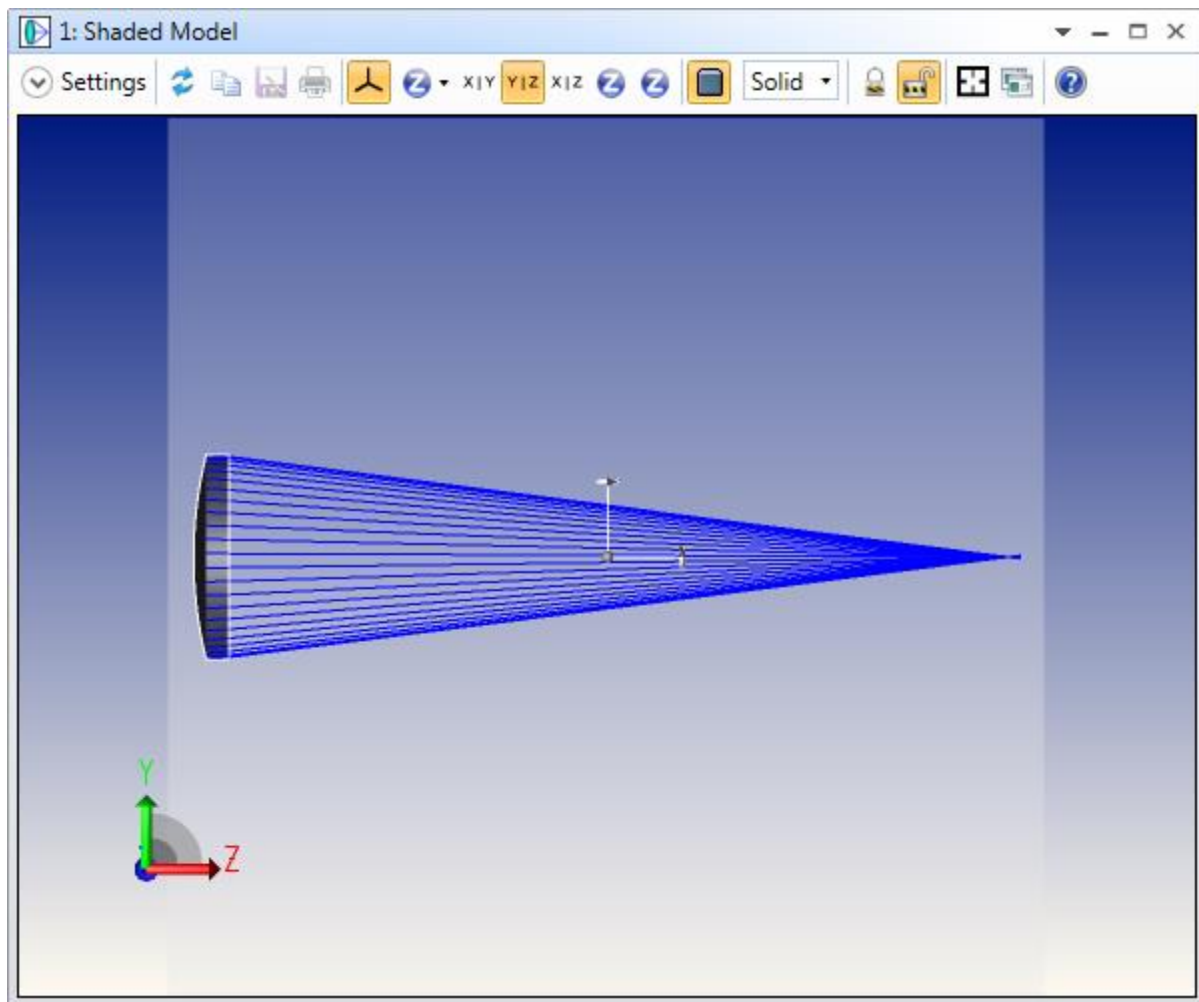
This uses the Shaded model in the YZ orientation, with a Y|Z slicing plane used to give a cross-section view as well as the solid model view.

Our goal is to design a scanning mirror that scans the focal spot through ±10° around a nominal 90° reflection angle.

First, note that the lens has been entered so that it is illuminated right up to the edge of its mechanical aperture. Since the lens will be held in a mount of some sort, we need to reduce the Entrance Pupil Diameter a little (since this is a catalog lens, the mechanical diameter is fixed). Open the System Explorer, and set the Aperture to be 18. Check the Units tab to see what the Lens Units are.

Next, click anywhere on surface 2 in the Lens Data Editor, to highlight surface2. Note that the thickness of this surface is controlled by a marginal ray height solve, and can be seen by the M following the thickness. This keeps the lens at paraxial focus. We will discuss solves more later. For now, simply press Insert after clicking on surface 2, to insert a new surface. Give this surface a thickness of 70 mm, and note that the thickness of the last surface automatically changes to keep this lens in focus:
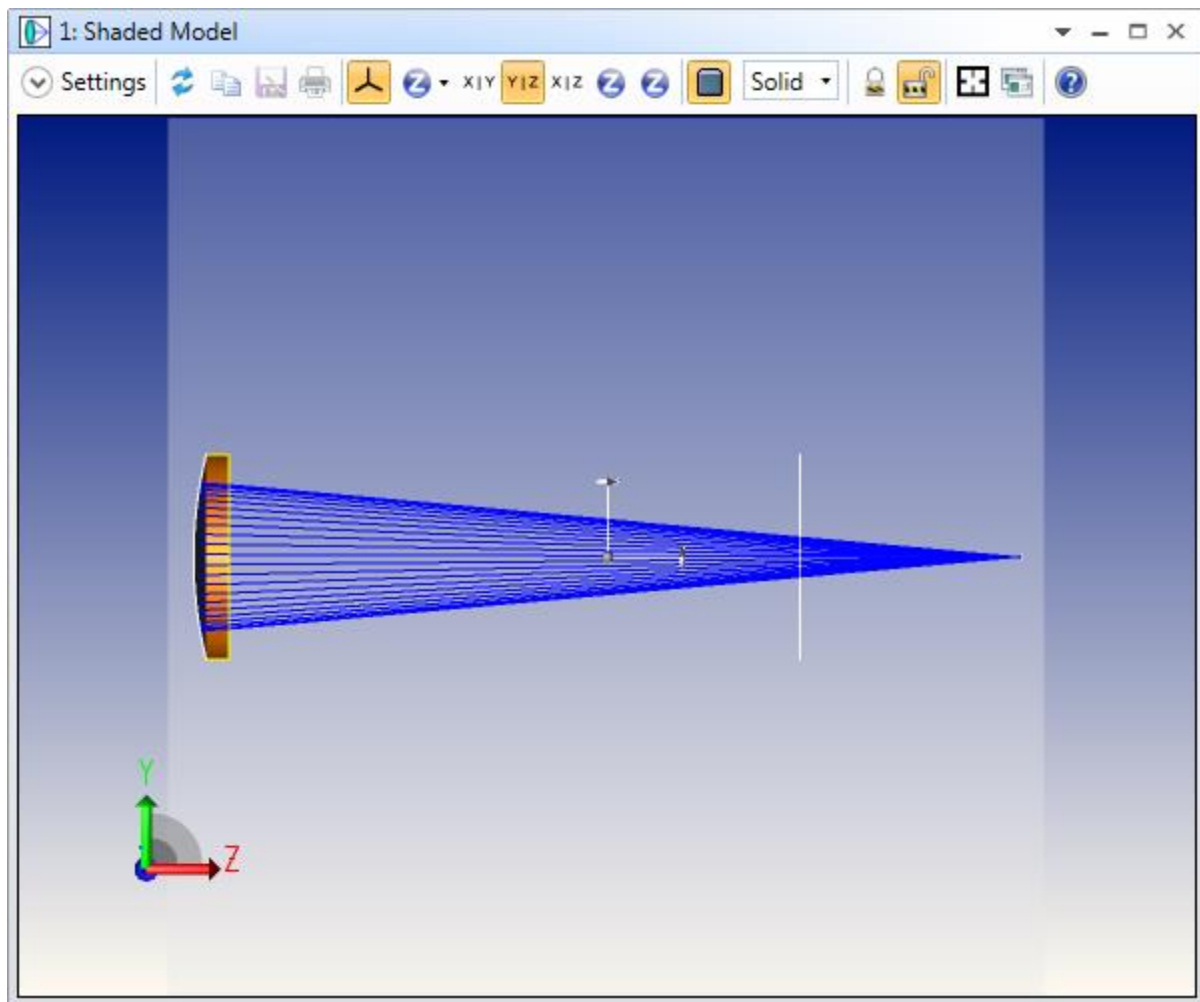
	Surf:Type		Comment	Radius	Thickness	Material	Coating	Semi-Diameter	Conic	TCE x 1E-6
0	OBJECT	Standard ▾		Infinity	Infinity			0.000	0.000	0.000
1	STOP (aper)	Standard ▾	32481	51.680	4.300	N-BK7		12.500 U	0.000	-
2		Standard ▾		Infinity	70.000			8.789	0.000	0.000
3	(aper)	Standard ▾		Infinity	27.165 M			12.500 U	0.000	0.000
4	IMAGE	Standard ▾		Infinity	-			0.328 U	0.000	0.000

We will now make surface 3 a 'Fold Mirror' which just reflects the light through some angle. Click on surface 3, and then click on the Add Fold Mirror icon in the Lens Data Editor toolbar:
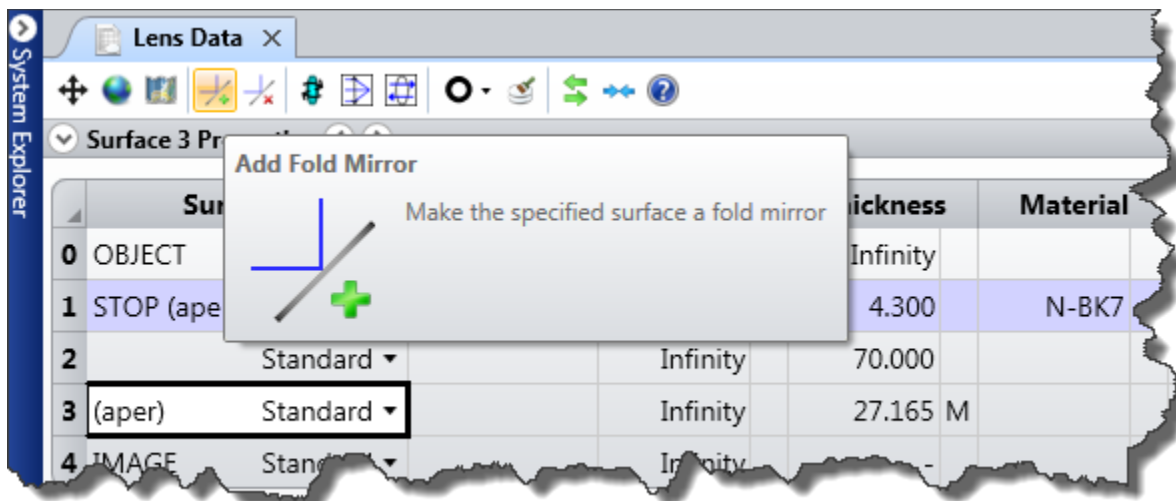
and configure it like so:
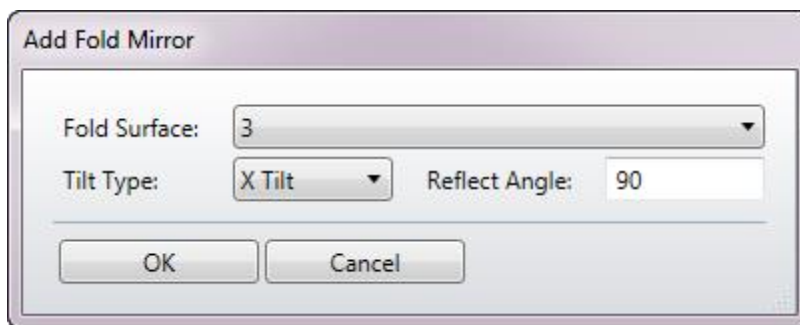
The Shaded Model will now show:
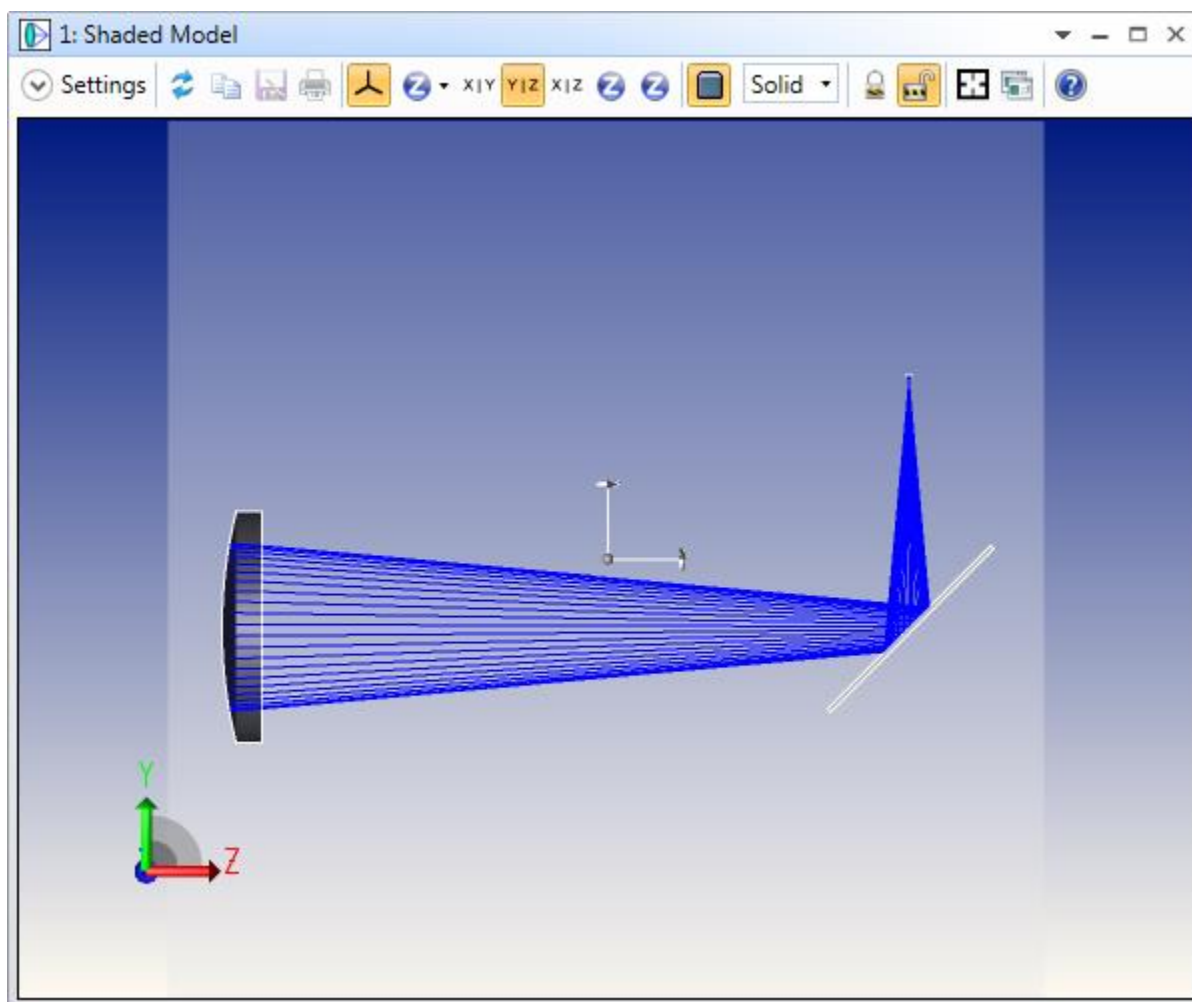
The Add Fold Mirror tool inserts two dummy surfaces, one before and one after the selected fold surface. The fold surface then has its material type set to MIRROR, which is a special status that tells OpticStudio that light should now reflect from this surface rather than refract into it. The two newly inserted adjacent surfaces are set to be Coordinate Breaks with the appropriate tilt angles. The second tilt angle is set as a pickup from the first tilt angle. Finally, all subsequent surface thicknesses and curvatures change sign to account for the new mirror.

Remember that light normally propagates in +Z (left to right) but after a mirror it obviously goes in the other direction. Using the Add Fold Mirror/Delete Fold Mirror tools automates all the tedious sign conversions.

We now want to scan this mirror through ±10°. We will initially tilt it through +10°, and then use multiple configurations to define multiple tilt angles. Click on surface 4 (which is the mirror surface now) and press the Tilt/Decenter icon in the Toolbar:
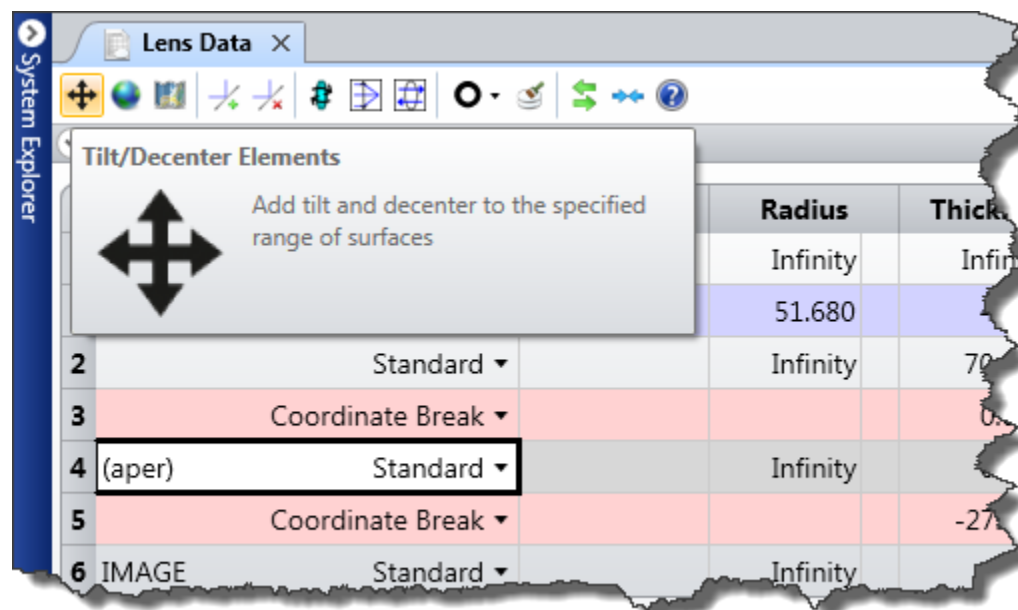
And configure as follows:
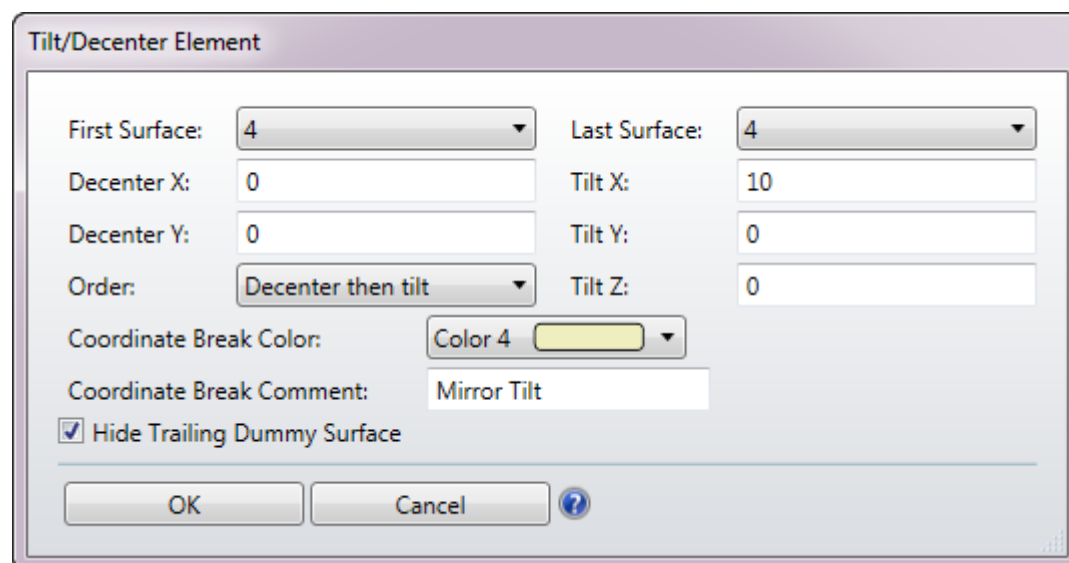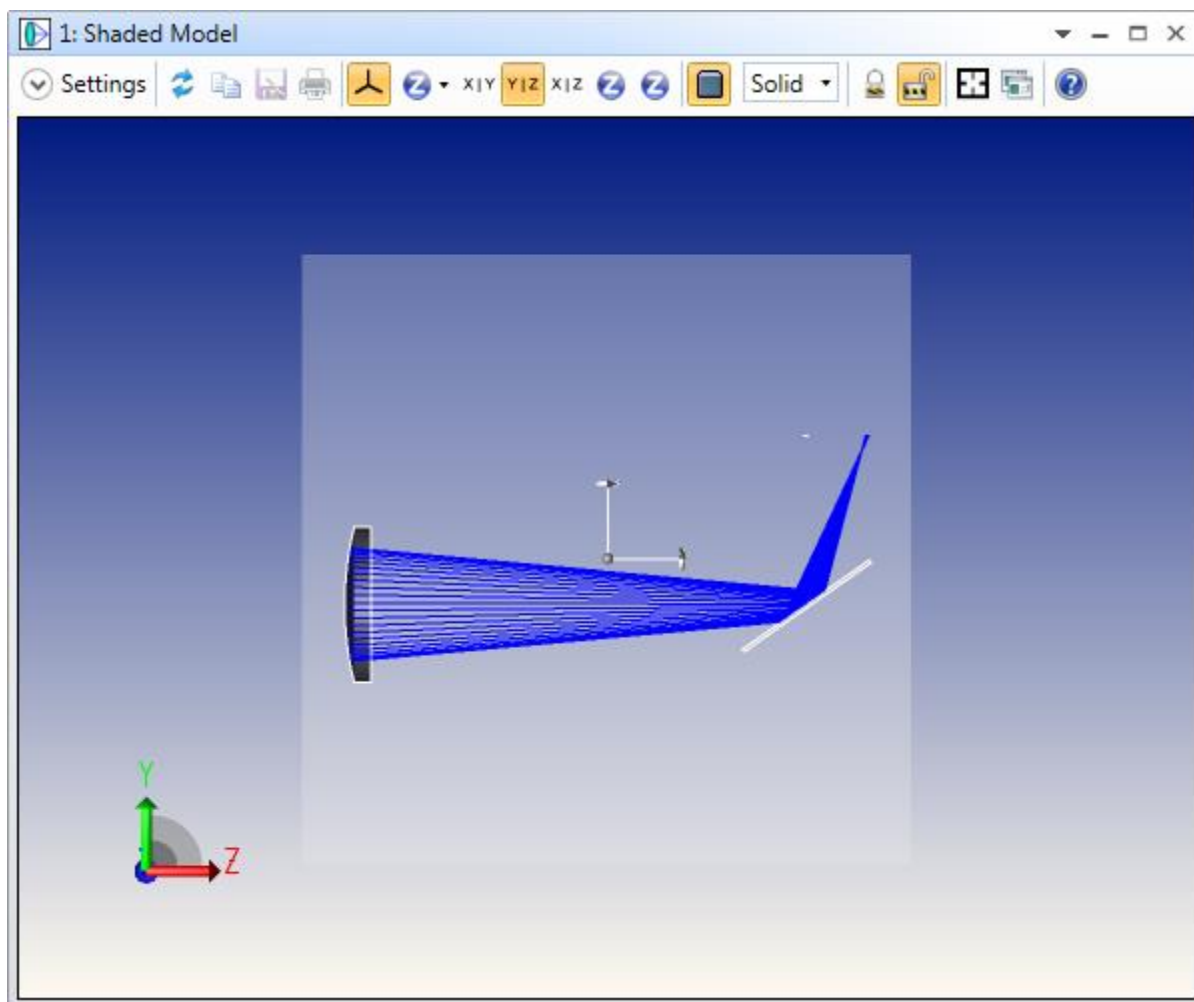
The Tilt/Decenter Element tool has operated just as it did in the earlier double-Gauss example, and it has added two more Coordinate Break surfaces, with pickups, that tilt the mirror surface. The two sets of Coordinate Breaks are 'nested', such that the set added by the Tilt/Decenter tool are inside the pair added by the Fold Mirror tool:

You can vary the +10° value by hand, and watch the Shaded Model, or click on the Optimize tab and select the Slider icon
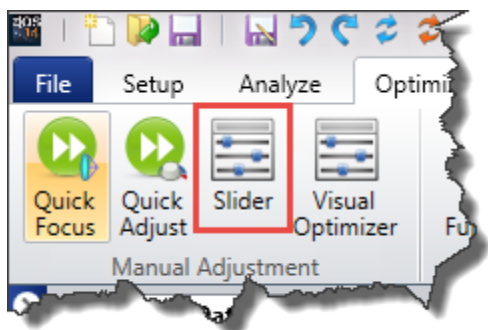
and configure it like so:
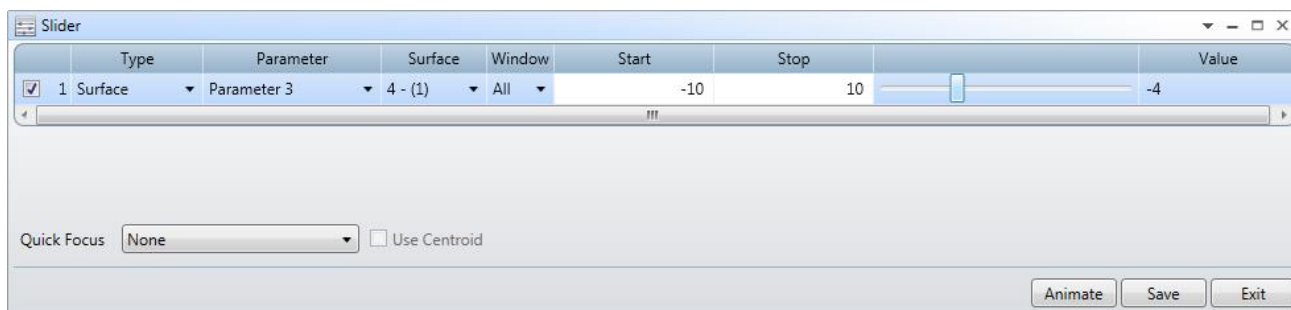
Press the 'Animate' button and all open windows will update as the Tilt About X (parameter 3) of surface 4 is scanned. You can also just drag the slider with the mouse. Now exit the slider, and reset the tilt in surface 4 to +10 degrees if needed.

Next, go to the Setup Tab, and press the Multiple Configuration icon in the Configurations group:
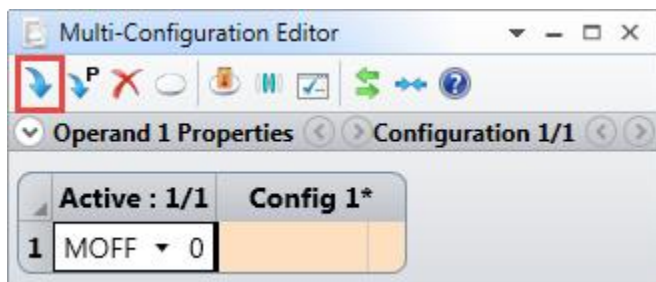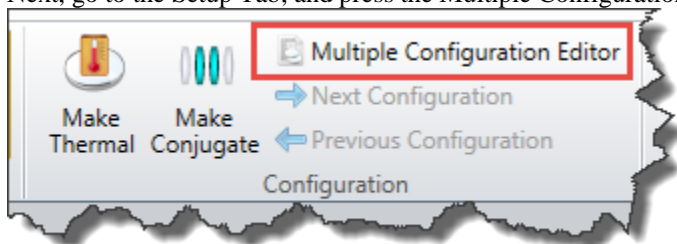
The lens currently has only one configuration, and the multiple-configuration operand MOFF ("Multiconfiguration OFF") is a placeholder that affects nothing, and allows you to enter comments in the editor if desired.

Note: The Multi-Configuration Editor is used to define everything that is different between configurations.

We will define 5 configurations, representing scan angles of +10°, +5°, 0°, -5° and -10° respectively. Click on the Multi-Configuration Editor, and press the Insert Configuration icon (highlighted above) four times. Alternatively, you can either right-mouse-click…Insert Configuration four times, or press the <Shift><Ctrl><Insert> keys simultaneously four times, that a total of five configurations results:
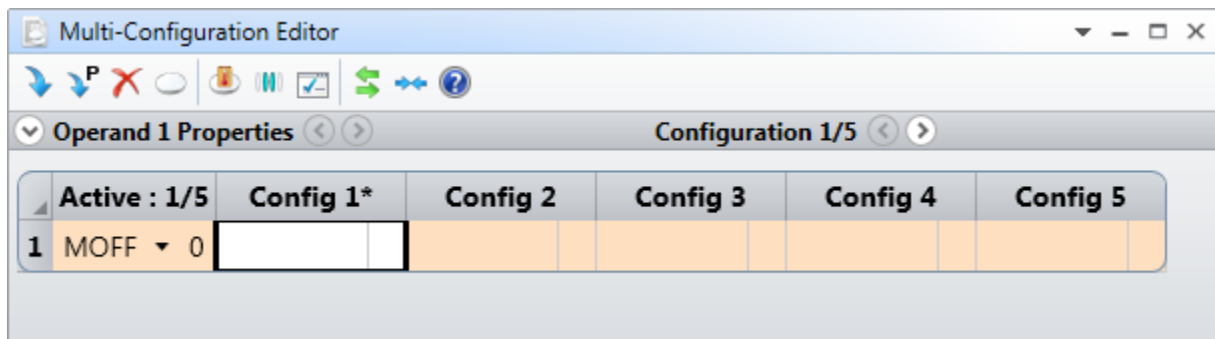
Each line in the Multi-Configuration Editor is an operand that acts on a parameter in the Lens Data Editor or some other System parameter and allows its value to be changed. Move the mouse over the 'MOFF' operand and double-left-click to edit the operand. All the multiple configuration operands OpticStudio supports can be selected from the drop-down list in the resulting dialog. Set it up like so:
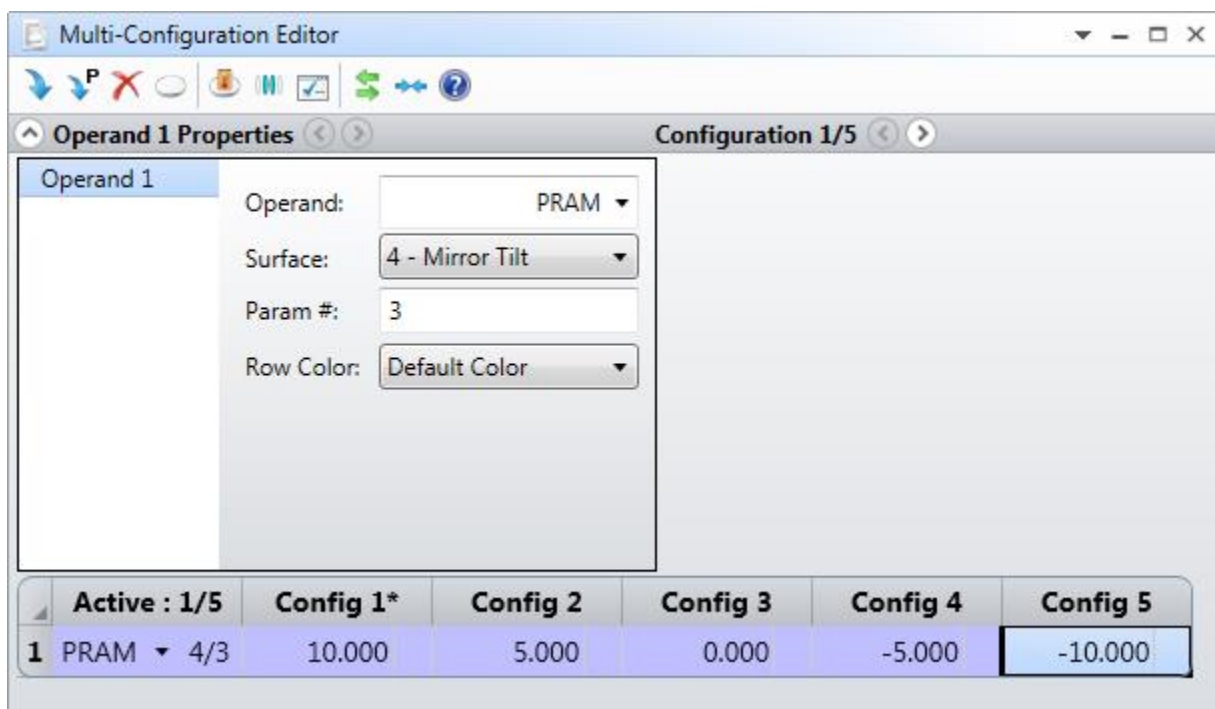
So that the PRAM operand picks up parameter data, in this case parameter 3 of surface 4, and changes it between 10, 5, 0, -5 and 10 degrees.

Use the <cntl>A keys to change the current configuration, and note that all open windows update to show the data for whatever configuration is current. Alternatively, use these controls:
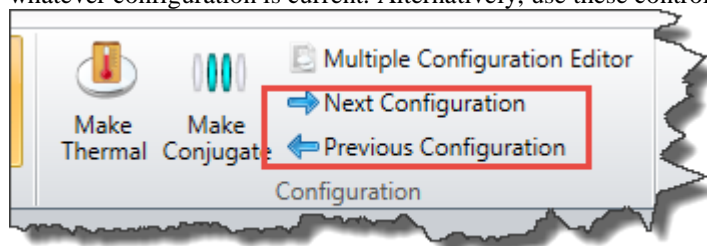
(Note that the Configuration group now appears on all tabs except the File tab as soon as the Multi-configuration Editor contains more than one configuration). You can also double-click on the header of each configuration's column to make it the active configuration.

Configure the Shaded Model like so:
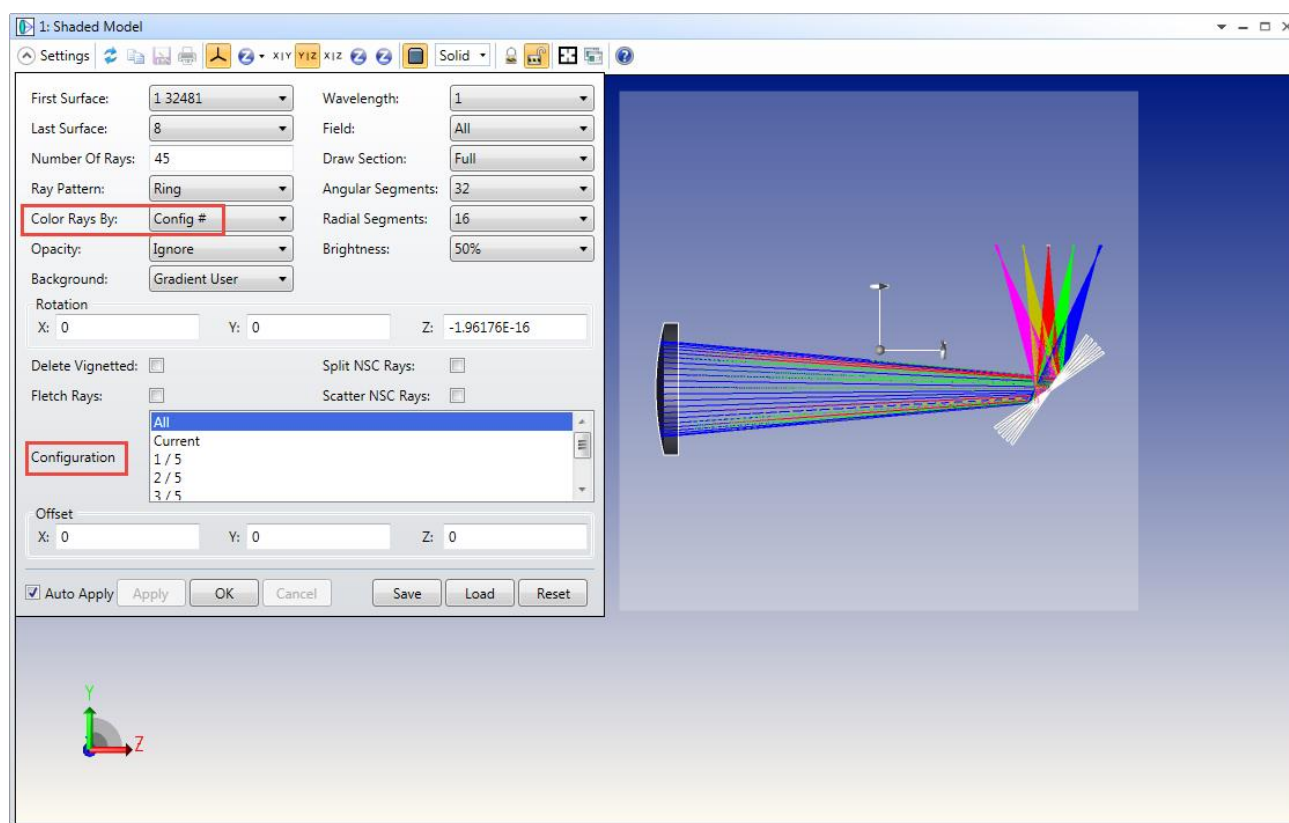
Note you can select any number of individual configurations to draw, like 1/5 and 4/5 by just selecting the desired configurations in the drop-down list.

Finally, click on the Analyze Tab…Spot Diagrams…Configuration Matrix
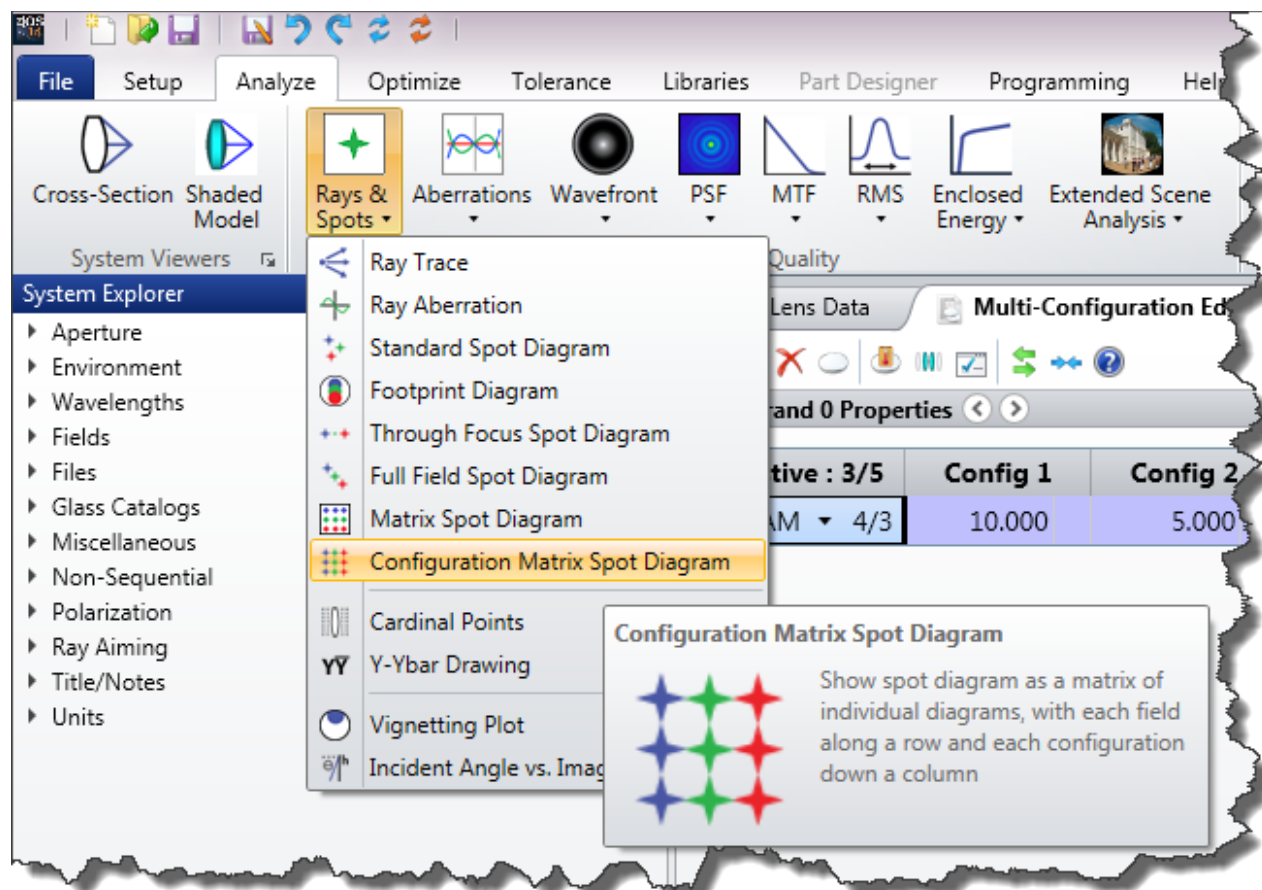
to see how the spot varies with scan angle:
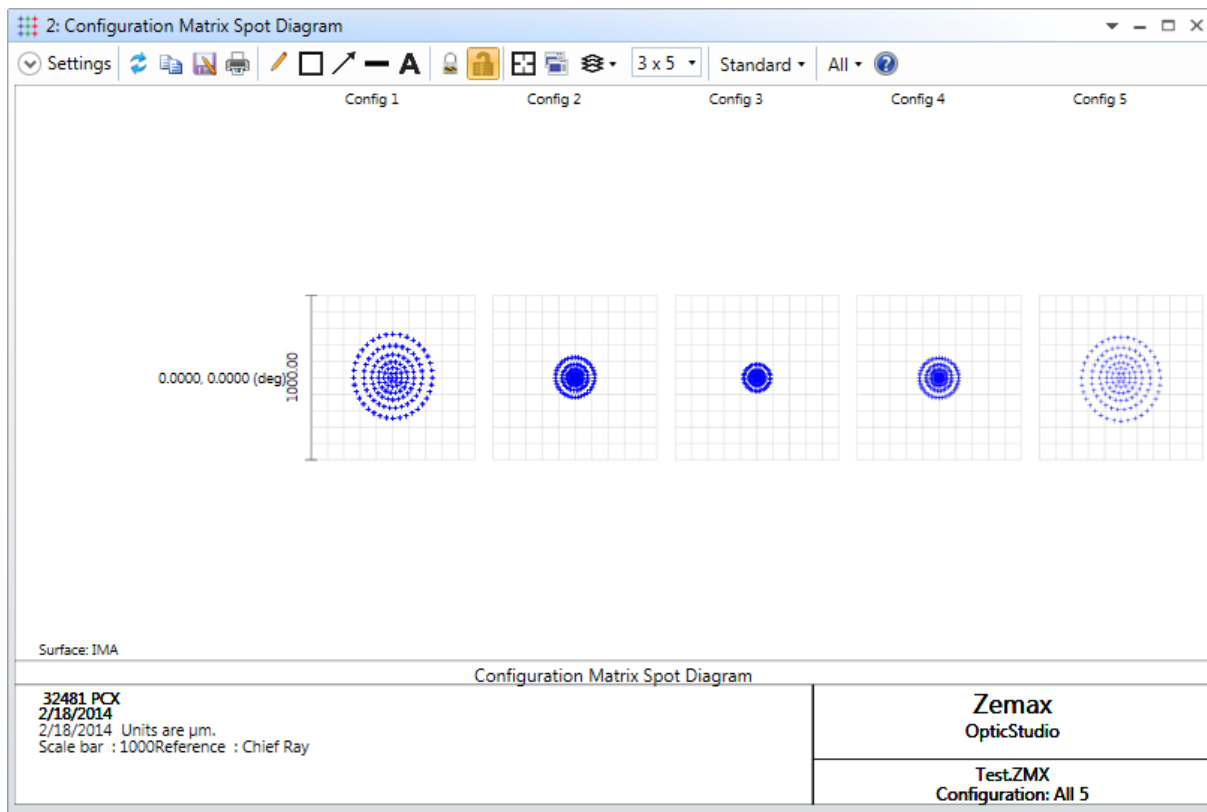
Finally, on the Lens Data Editor Toolbar, choose Apertures…Convert Semi-Diameters to Maximum Apertures
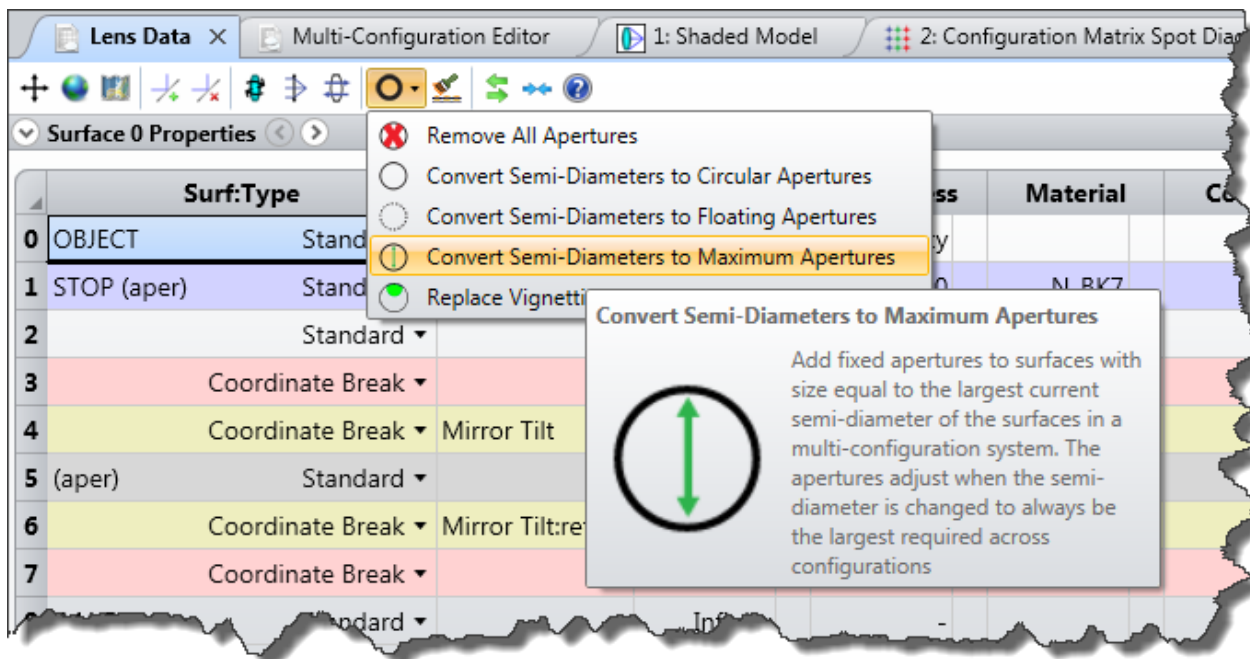
This will set the semi-diameters of all surfaces to the largest of whichever configuration needs it:
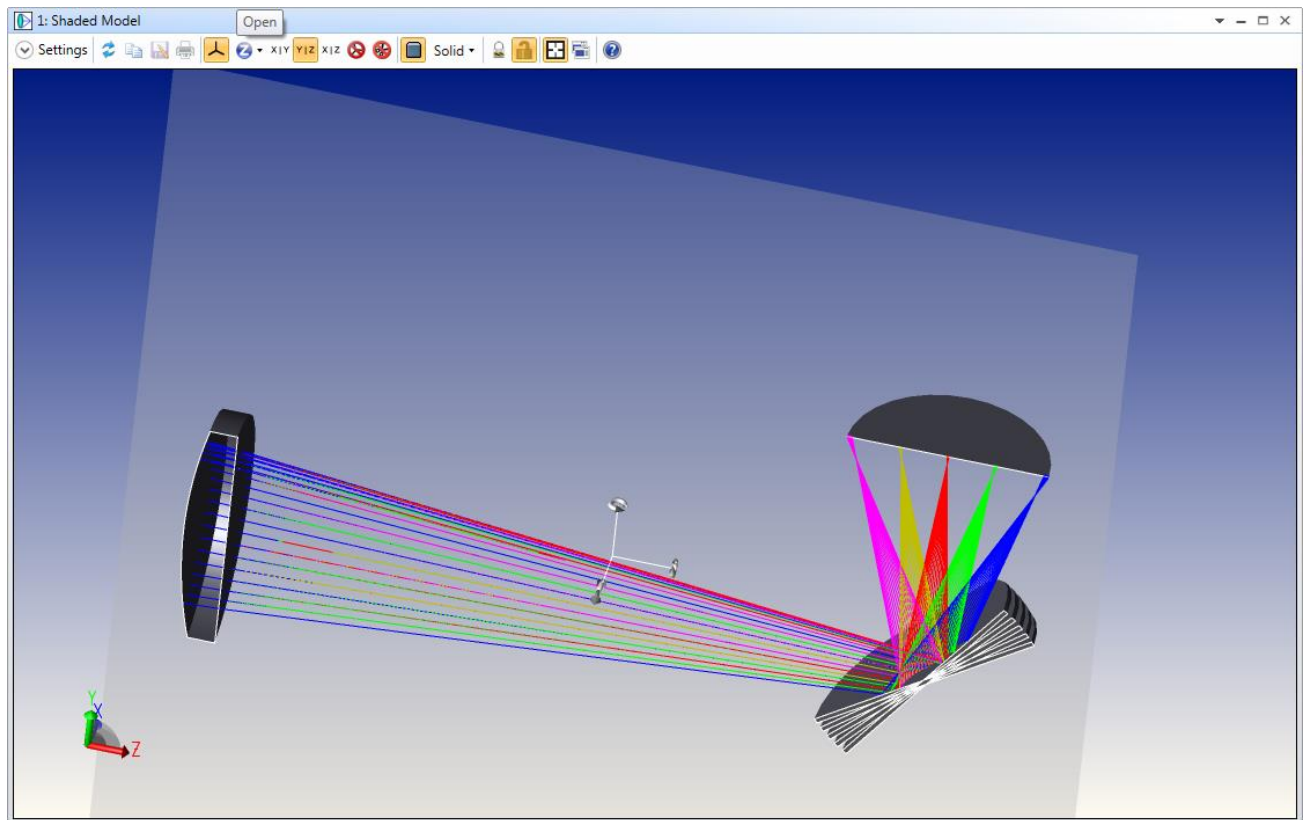
Tutorial 4: Exporting To Mechanical CAD Packages

At some point in your lens design, you may want to export the lens design into a mechanical design package so that you or your mechanical engineering colleagues can design lens mounts and other opto-mechanical components alongside the lenses and rays. This is easily achieved using the Export group in the File tab:
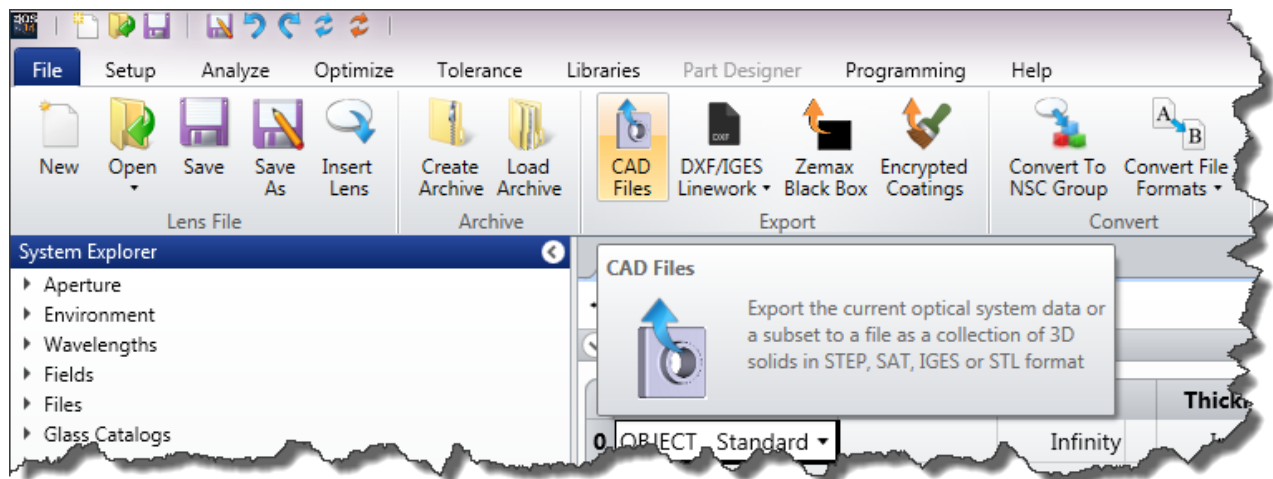
Using the file designed in Turorial 3, export it like so:
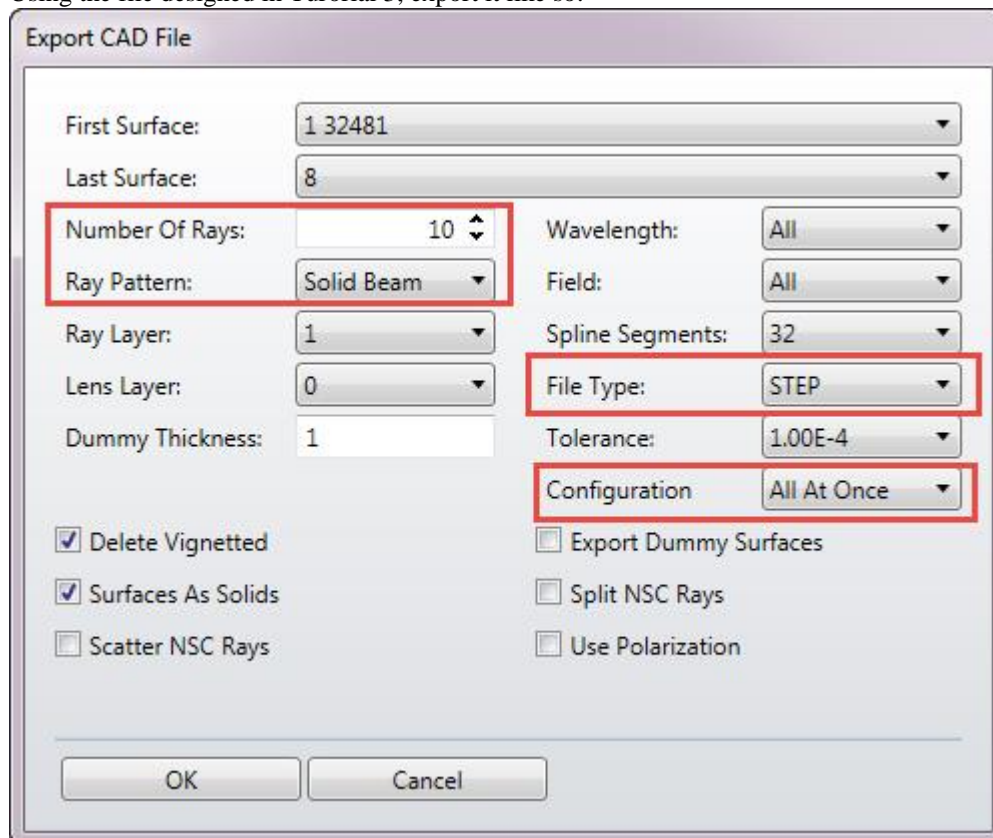
And load the file produced into your 3D CAD package of choice:
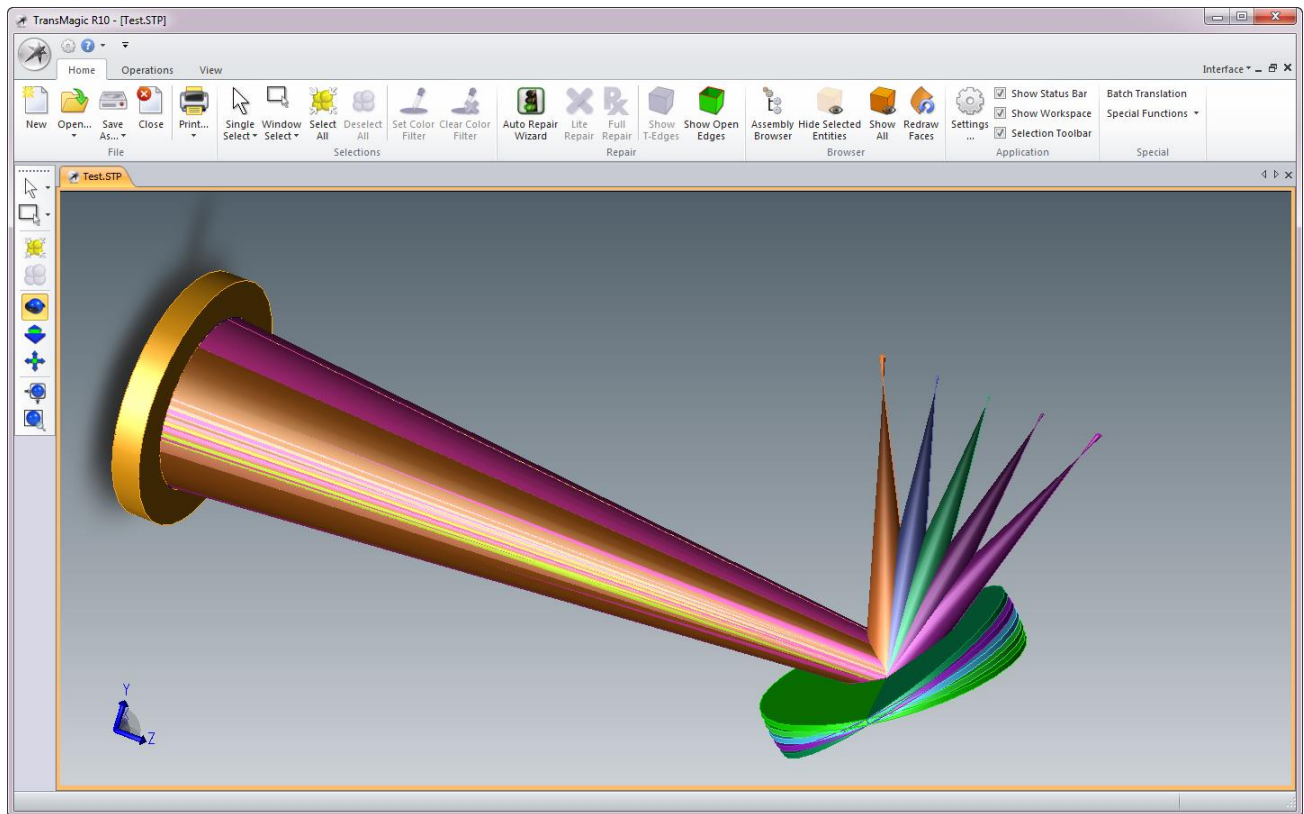
Note: If you are using SolidWorks™, and do not see the exported rays, visit the OpticStudio User's Knowledge Base at www.zemax.com/kb and search for "SolidWorks" to get advice on how to set SolidWorks' importer options.

This lets you see easily the range of motion of the mirror, the envelope of the rays, and the optical components. Multiple configurations can be exported as separate files, as separate layers of the same file, or 'all at once' as done here. Importing CAD objects is discussed in the non-sequential ray-tracing section of this Guide.

Tutorial 5: Optimization

So far, we have loaded a lens file, and used Analysis features to look at the performance of the system. Optimization takes this to the next level: OpticStudio will actively work with you to obtain the best possible system performance, and not simply report what the system performance is.

Note: Even if you intend to use only the non-sequential mode of OpticStudio you should work through this example, as the concepts of optimization are the same in non-sequential ray-tracing as in sequential.

Optimization is a three step process:
- First, a basic lens design is entered which has the correct field of view, wavelength, system aperture, number of surfaces, etc. This system should be traceable without error.
- Next, some parameters in the Editor are defined as variables. This means that OpticStudio can change the value of these variable parameters in order to better meet the design specification.
- Last, the design specification is expressed as a series of design goals called a merit function. The merit function ultimately is expressed as a single number, and the closer to zero this number is, the closer the design is to your desired performance.

Optimization then changes the values of the defined variable parameters so that the merit function is reduced to its minimum value. OpticStudio contains several different optimization algorithms, two local optimizers and two global optimizers. In this example we will use the Damped Least Squares local optimizer, and the Hammer global optimizer. Also, this three-step optimization process may be repeated several times during the design process. After optimization, the system performance may still not be as required, and more lenses may be needed, or you may choose to make some surfaces aspheric and repeat the optimization with the aspheric parameters set as variables.

Tutorial 5.1: The Lens Specification

Here is the specification of the lens we are to design:

```
Design a cemented doublet that works at f/5 over the visible region of the
spectrum. The field of view is 10° full field of view, and the object is a very
long distance away from the lens. The lens aperture is 25 mm entrance pupil
diameter, and the lens must be at least 2 mm wider in diameter than the optical
beam to allow for mounting clearance.
```

Tutorial 5.2: Entering the Basic System in the System Explorer

Press File…New to clear OpticStudio and start a new design. We will start by defining the incoming light, by its aperture, wavelength, and field of view. All of this is defined in the System Explorer sidebar.

The system aperture has been defined as 25 mm entrance pupil diameter, with at least a 27 mm mechanical diameter to allow for mounting of the lens. In the System Explorer Aperture Group, enter the Entrance Pupil Diameter as 25:
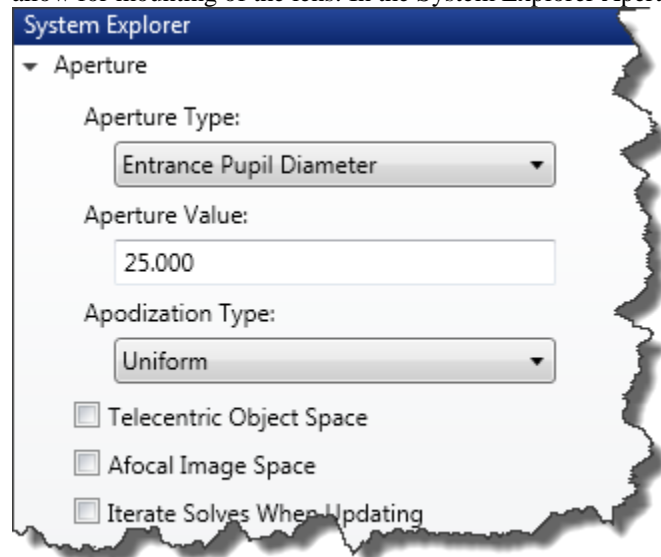
Lens units are millimeters by default, which you can confirm by clicking on the Units tab of the dialog box. We also have a requirement for the lens to be mechanically 2 mm larger in diameter than its working aperture, so click on the Miscellaneous tab and enter a 1 mm semi-diameter margin like so:
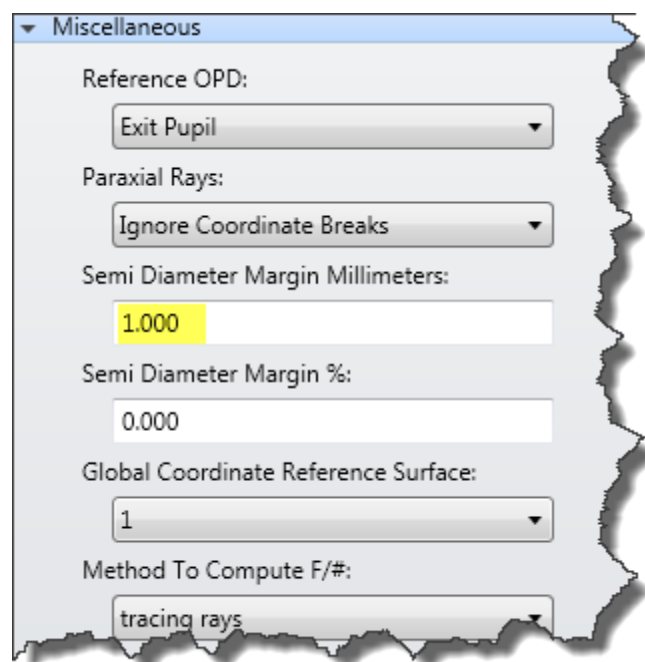
Note that OpticStudio works with semi-diameters, not diameters, so a 1 mm semi-diameter margin gives a 2 mm diameter margin.

We will now define the wavelength of the light. This specification is a little vague on this important system parameter. For example, what is the 'visible spectrum', and at exactly what wavelength should the system be f/5? We will assume that the 'visible' region is that covered by the F, d and C spectroscopic lines. This is a very common assumption in

visible system design. We will further assume that the d-line, being the central wavelength, should be the wavelength at which the f/# of the lens is defined. We will therefore enter the System wavelengths as follows
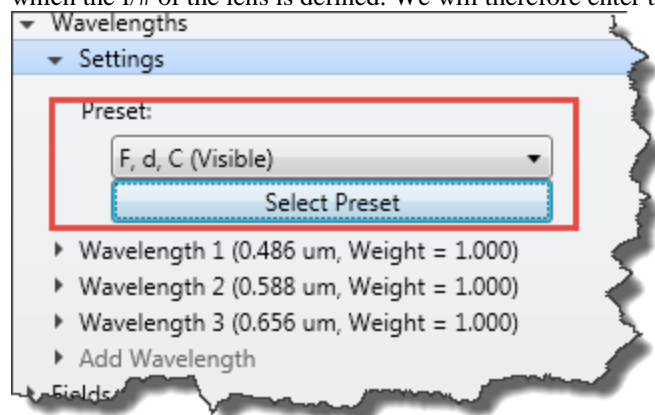
In the drop-down list next to the 'Select' button, choose the spectrum "F, d, C (visible)" and press the Select button to copy this spectrum into the Wavelength Data dialog. This sets three wavelengths, in ascending order, and sets wavelength #2 as the primary wavelength.

Note: In real life, any time you find yourself making assumptions about what specifications mean, always refer back to the customer! Part of your QA process should be to do a point-by-point comparison of each specified parameter, how it has been entered into OpticStudio and how it will be tested in the built system.

Last, we will define the field of view, which is 10° full field of view, hence 5° half-field. In the Field section, click on Add Field:
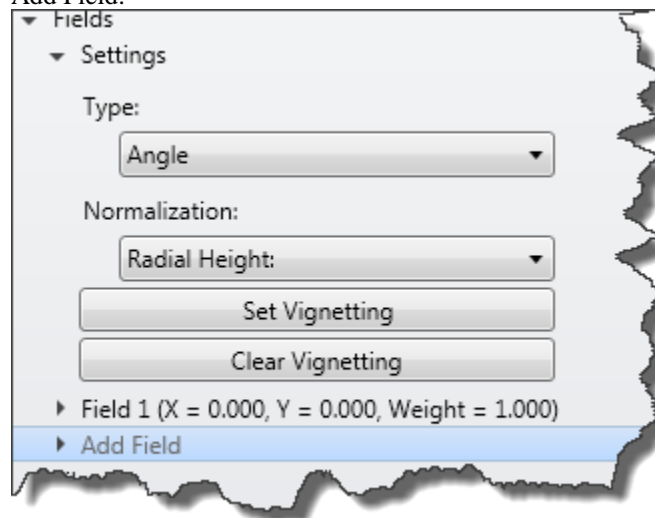
And enter the new field point like so:
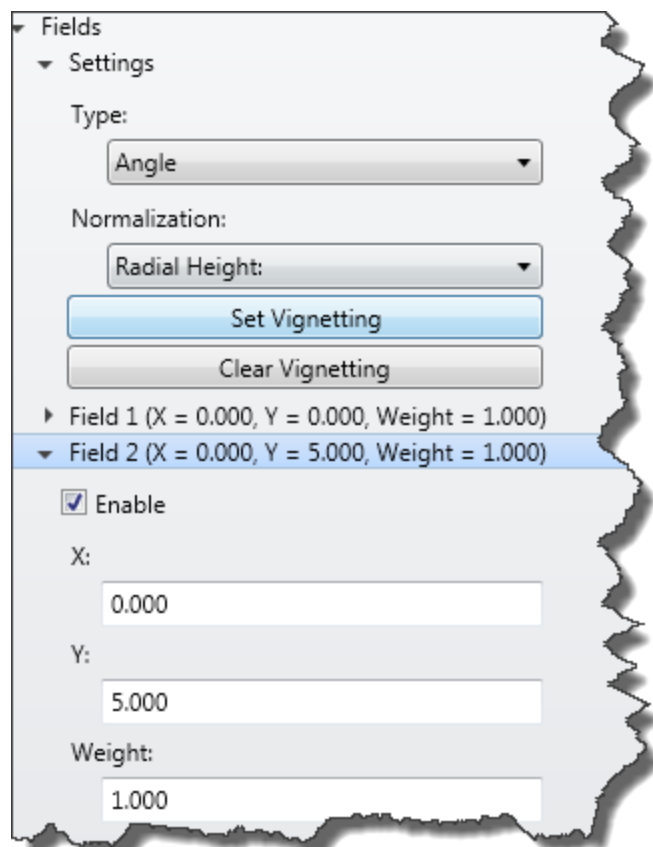
Alternatively, just double-click on the 'Fields' line at the top of this section, and enter the data though a standard dialog box:

Note: As the lens is rotationally symmetric we do not need to specify a field point at y = -5°, or at x = 5° or -5°. Always define your field points in +y only unless you specifically require a non-rotationally-symmetric lens system. We have now entered everything we need about the light coming into the lens: its diameter, wavelength and field of view.

Tutorial 5.3: Entering the Basic System in the Lens Data Editor

We now need to enter the first-guess data for the optical surfaces in this lens system. As we know we will be designing a cemented doublet, we know we will need a total of six surfaces: the OBJECT surface, STOP surface, the front, middle and rear doublet lens surfaces and the IMAGE surface. Go to the Setup Tab, and click the Lens Data icon in the Editor group. Click on the IMAGE surface in the Lens Data Editor, and press the Insert key 3 times to insert the correct number of surfaces, and enter the following data:

	Surf:Type		Comment	Radius	Thickness	Material	Coating	Semi-Diameter	Conic	TCE x 1E-6
0	OBJECT	Standard ▾		Infinity	Infinity			Infinity	0.000	0.000
1	STOP	Standard ▾		Infinity	50.000			12.500	0.000	0.000
2		Standard ▾	front	Infinity	5.000	N-BK7		17.874	0.000	-
3		Standard ▾	middle	Infinity	5.000	F2		18.163	0.000	-
4		Standard ▾	rear	Infinity	50.000			18.433	0.000	0.000
5	IMAGE	Standard ▾		Infinity	-			21.807	0.000	0.000

Note: If you are using the demonstration version, OpticStudio will display fewer decimal places. This does not affect the accuracy of the calculations or results.

Because the specification says that the OBJECT scene is 'a very long distance' away from the lens we have set the OBJECT surface thickness to 'Infinity'. You do this by typing the word "Infinity" or just the letter "i" (without the quotation marks) in the editor cell.

Do not enter anything in the semi-diameter data column. OpticStudio will work this out for you, and include the requested margin so that the lens is larger than the incoming beam. Use the Cross-section layout (in the Setup and Analyze tabs) to show the design so far:
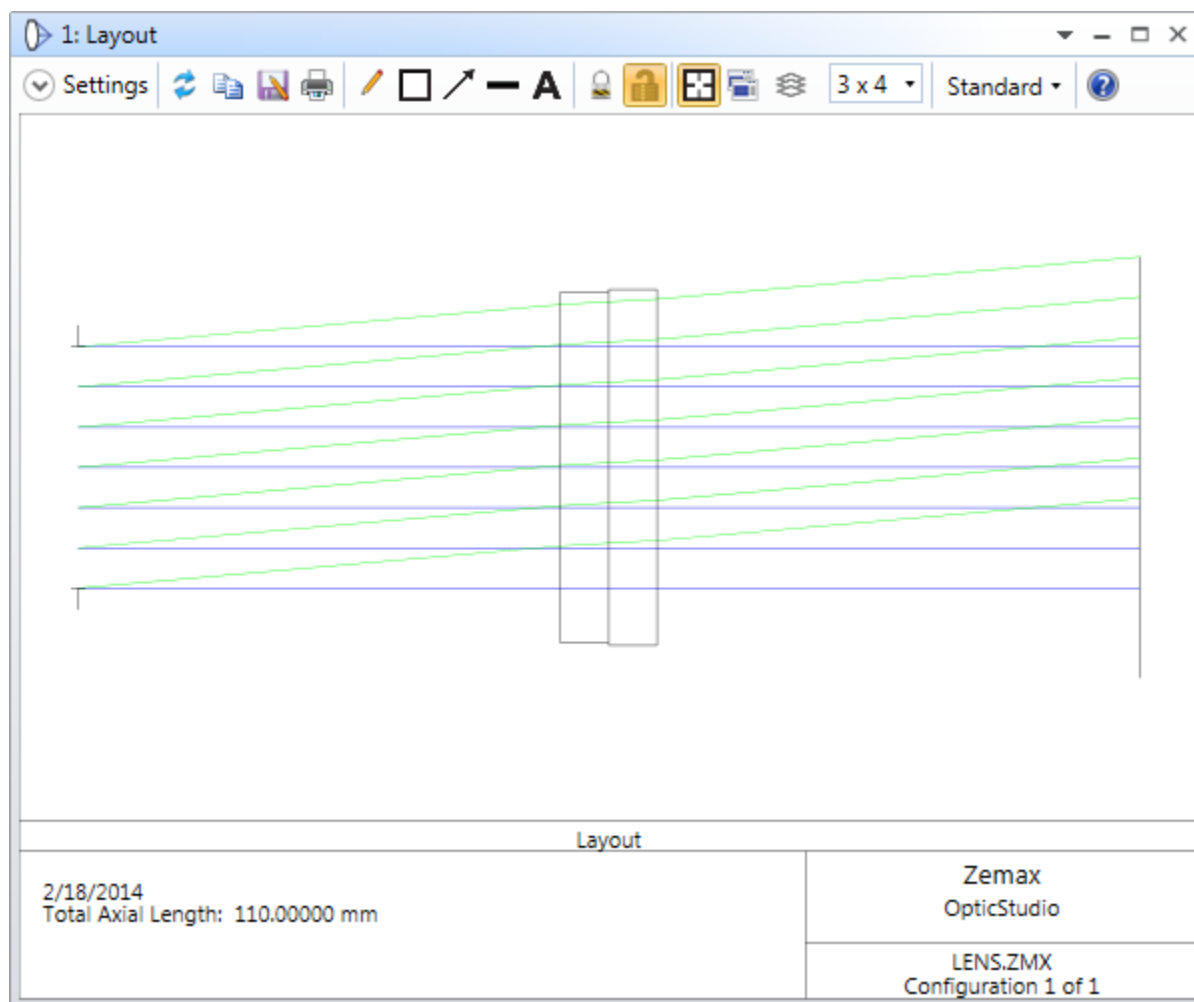
Now our system has a requirement that it must be f/5. There is a simple way to achieve this, click the cell to the right of the radius of curvature cell of the last lens surface, and choose an f-number solve like so:
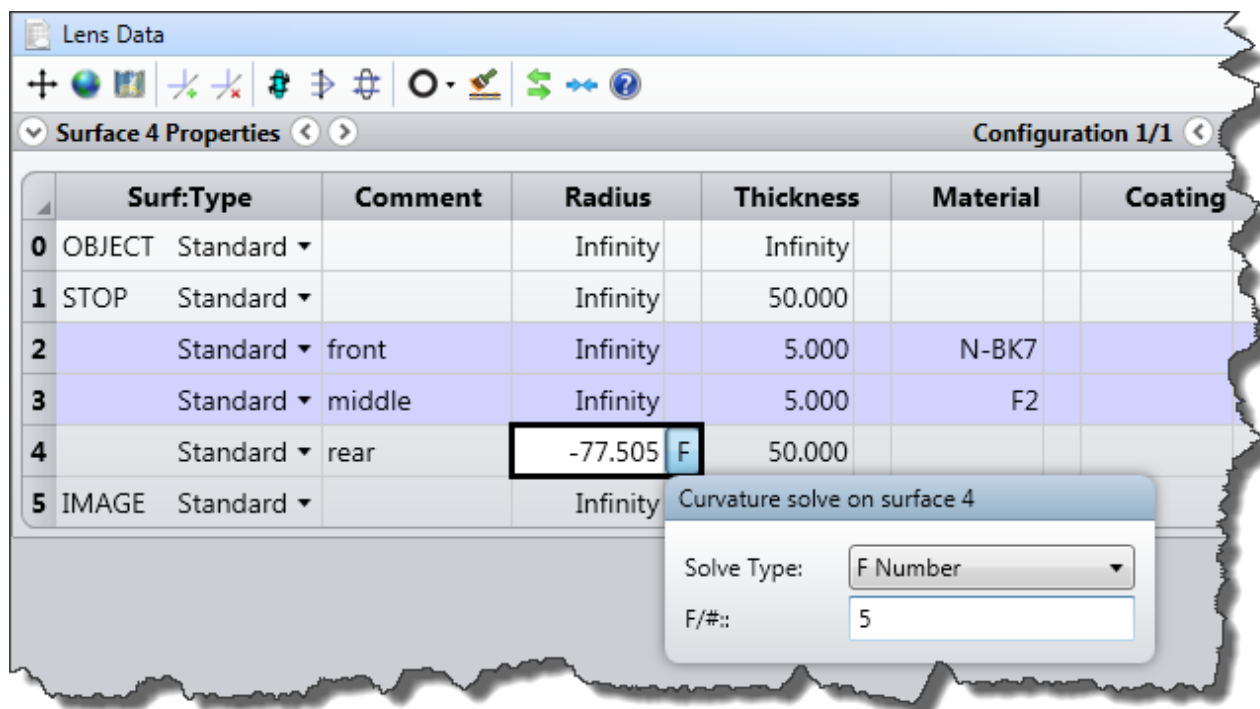
OpticStudio will immediately compute the radius of curvature that yields an f/5 cone of light:
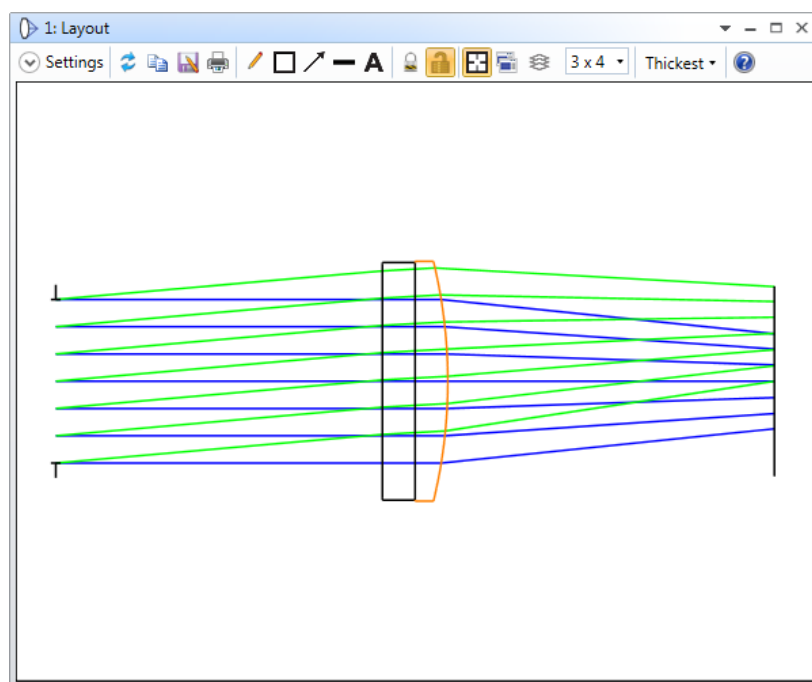
Try altering the radii of curvature of the other two surfaces, and you will see that the f/# solve automatically updates to enforce the condition that the lens be f/5. A solve is the most efficient way to enforce a system constraint.

Note: Read the "Solves" section of the Technical Reference in its entirety. A solid understanding, and use, of solves is one of the hallmarks of the professional lens designer!

Now we will bring the lens into focus. Go to the Optimize tab, and in the Manual Optimization tab choose the Quick Focus icon:
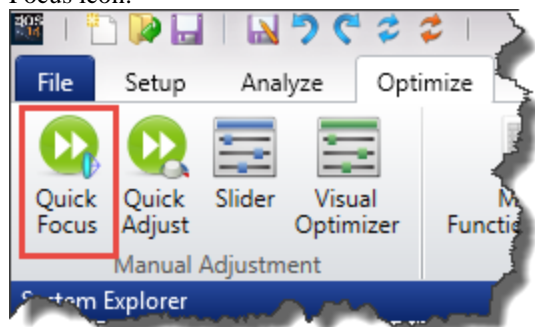
and set it up like so:
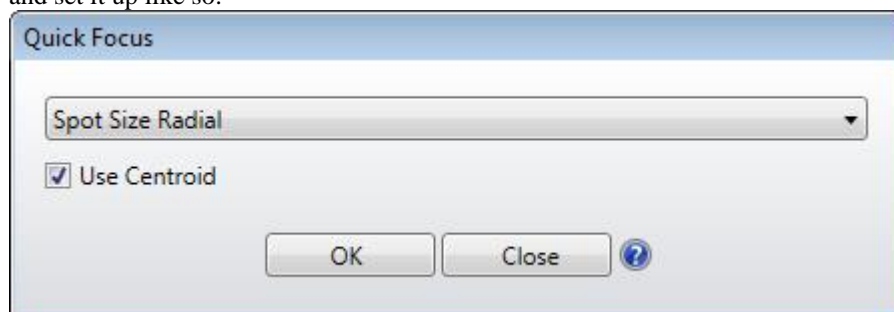
Update the cross-section window to see the final 'basic setup':
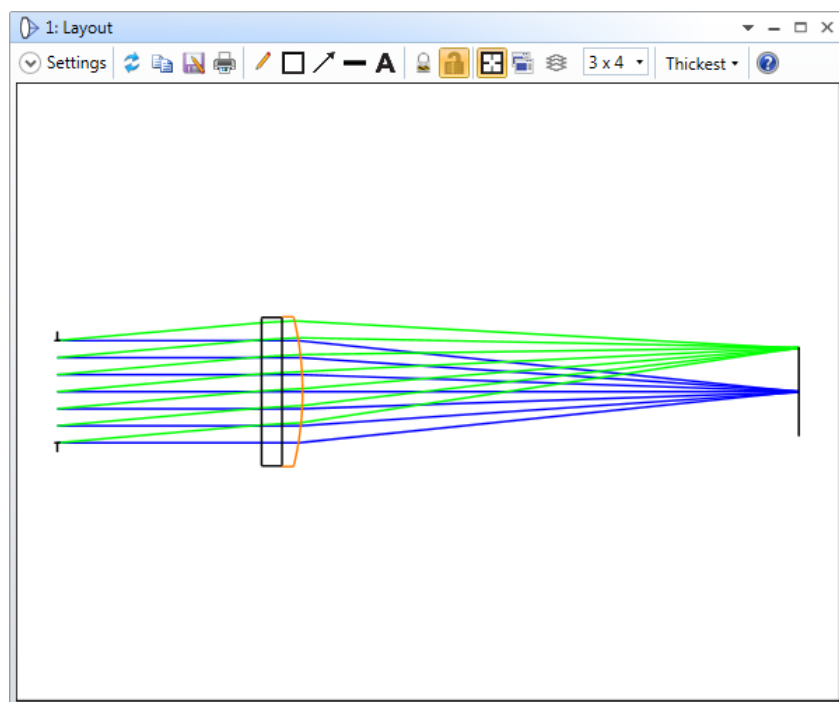
So now we have an f/5 lens, operating over the visible with a 5 degree field of view. Open a spot diagram by clicking on the Standard Spot Diagram in the Rays and Spots menu of the Analyze tab:
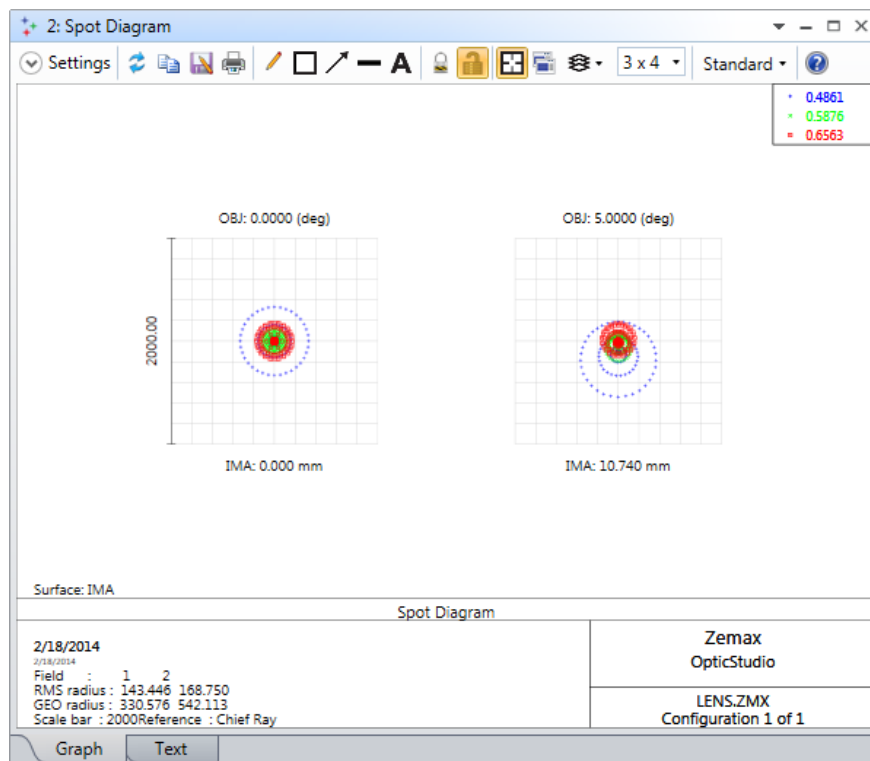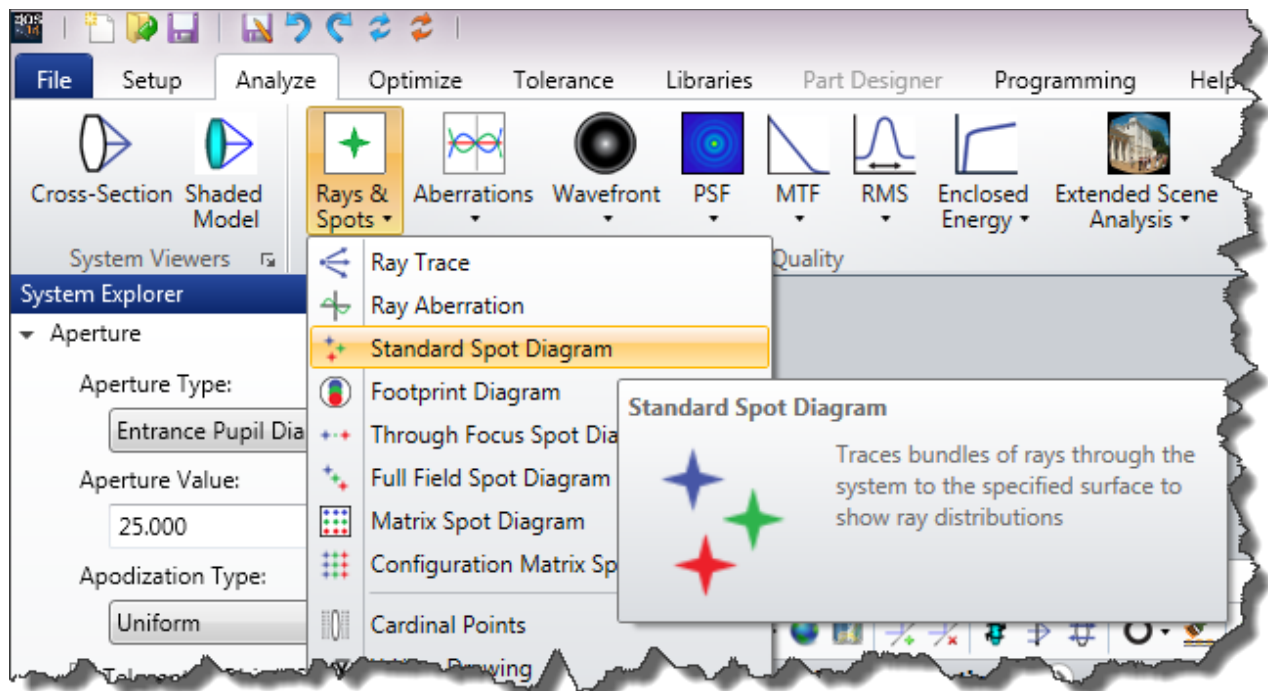
The RMS spot radius is 143µ on axis, and about 169µ at the 5° field point. Check that you get the same data as shown here, and if not go back through the exercise step by step to make sure your system is correctly set up.

Note: Finally, click on Setup Tab…System Check in the 'Diagnostics' group. This invaluable utility checks your file for the most common setup faults. Although not every possible fault can be caught by such a utility, anything it does report should be checked, and anything classed as an 'Error' must be rectified before proceeding.

Then click on File…Save As, to save the file as 'basic setup.zmx'.

Setting Variables

Our basic system setup is certainly an f/5 lens that meets the specification of aperture, wavelength and field of view, but it is not necessarily the best possible lens for the job. In fact, with only one curved surface, it is highly unlikely to be the best possible lens for the job! We are now going to optimize the lens to get the best possible performance.

First we will tell OpticStudio what it may change. We do this by clicking on the cell to the right of the parameter we want, and selecting the 'variable' solve:
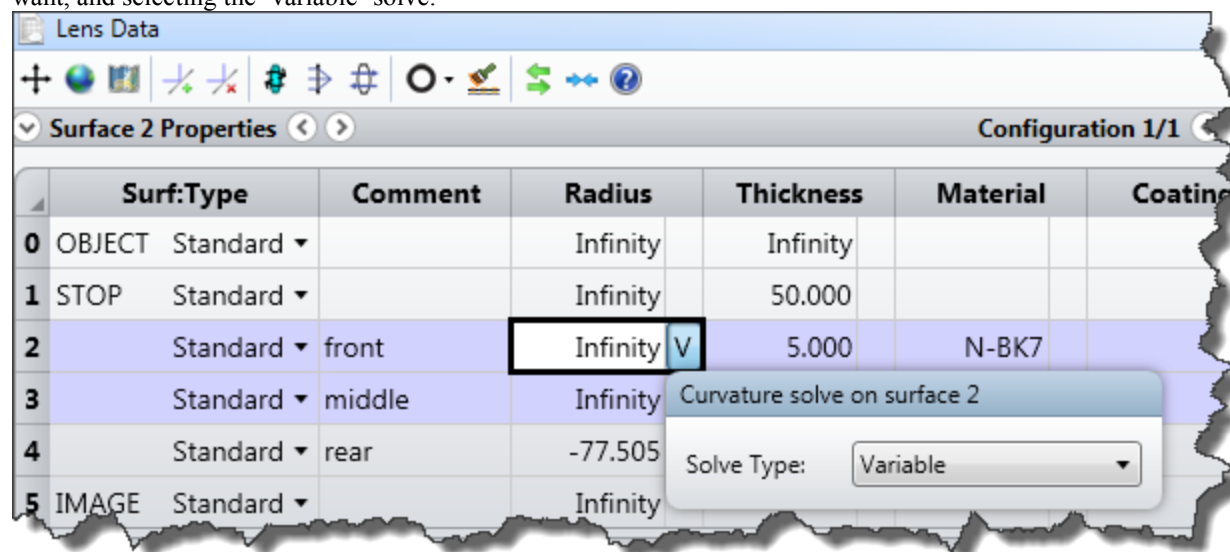
Or we can highlight the cells we want OpticStudio to change the values of, and use the keyboard shortcut <Cntl>Z (press and hold the Cntl button, and then press the z button on the keyboard):
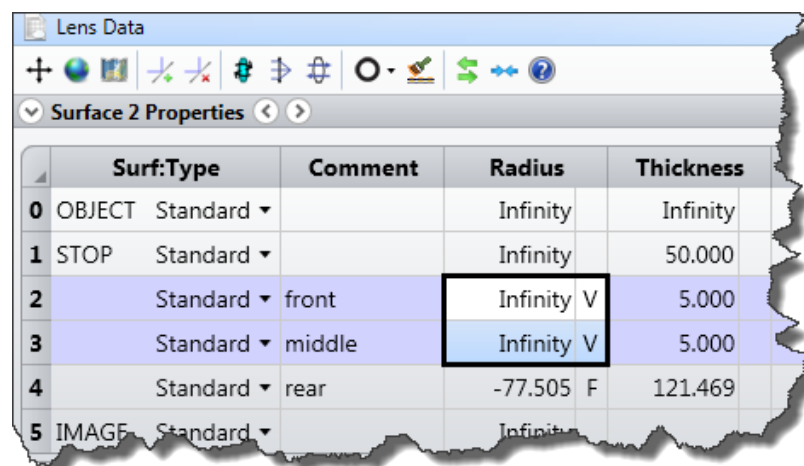
Or, in the Automatic Optimization group of the Optimize tab, we can use set all radii variable/set all thicknesses variable:

To set a total of six variables:

The status flag V indicates variables that OpticStudio may change the values of, just as the F flag means that the rear surface's radius of curvature is set by an f/# solve. As OpticStudio modifies the values of the variables, the f/# solve will automatically update to maintain the lens at f/5.

Defining the Merit Function

Next we will build the merit function for this design. In the Optimize tab, click on Optimization Wizard in the Automatic Optimization group:
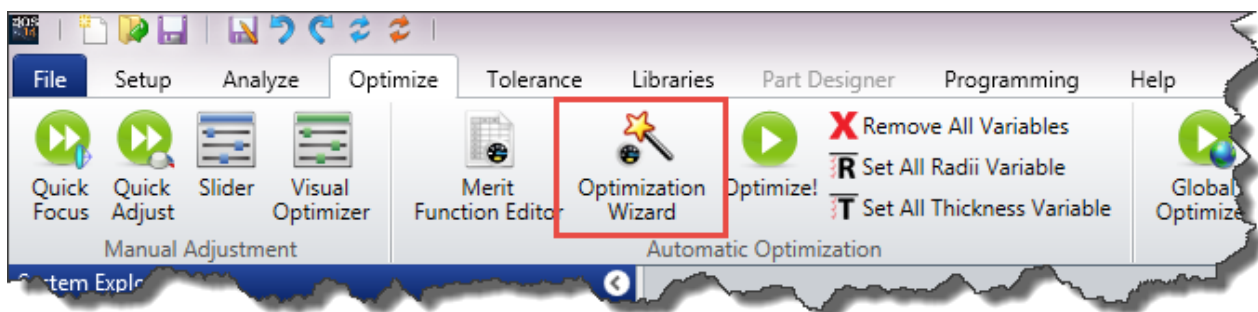
This will open the Merit Function Editor, and also activate the Optimization Wizard in the Property Inspector area of the Editor. Because this is a focal system, we want the smallest RMS spot radius, choose RMS Spot Radius, relative to the centroid, and set the number of rings to 4 (we will discuss this in more detail later, for now just make these changes) and press the OK button to close the Optimization Wizard:
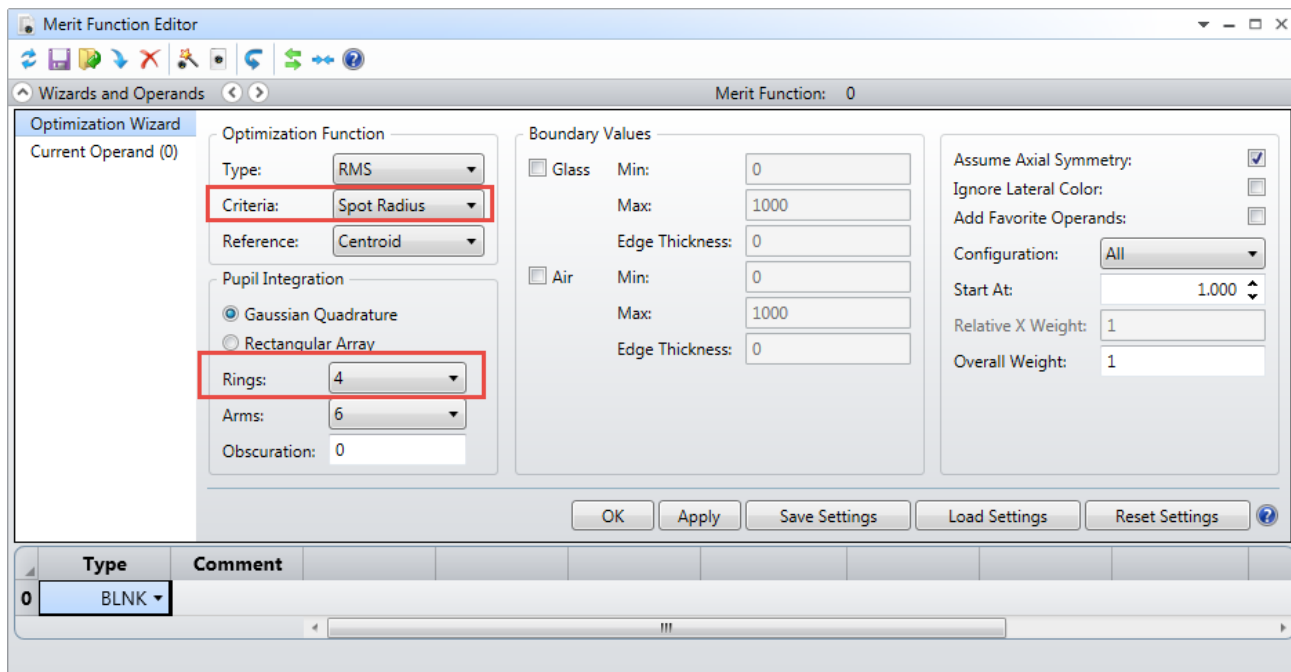
OpticStudio will then write out a merit function like so:

Each row in the Merit Function Editor contains an operand, which computes some value. The TRAC operand, for example, computes the radial point at which a specified ray lands on the image plane, relative to the average of all rays from that field point. Note that each TRAC operand traces a ray defined by its wavelength number, and its (Hx, Hy, Px, Py) normalized coordinates. Different operands will take different arguments, and the names of the arguments are given in the header row of the Editor.

Each operand that computes a value returns that value in the 'Value' column of the editor. The operand is also given a target value to achieve, and a weight. The merit function value is then computed as:

$$MF^2 = \frac{\sum W_i(V_i - T_i)^2}{\sum W_i}$$

where W_i is the weight of the ith operand, V_i is its computed value and T_i is its target value, and the summation is over all the operands in the merit function. As the computed values of the operands move towards their target values, the merit function value approaches zero. Because the difference between the target and actual values of each operand is squared, any deviation from the target value yields an increasingly positive value of the merit function.

Note: The goal of the optimizer is to reduce the merit function to zero, or as close as possible, by adjusting the values of the variable parameters in the Lens Data Editor.

Optimizing the Lens

Now that we have defined the variables and the merit function, click the Optimize! icon in the Automatic Optimization group
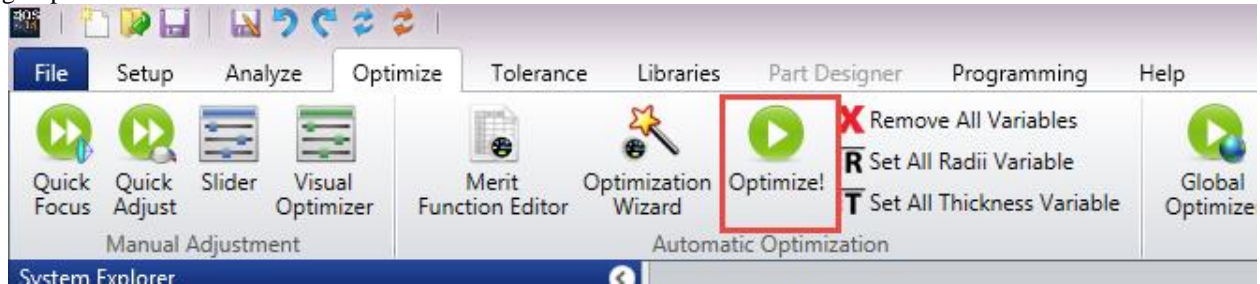
And then press the Automatic button. Note that the optimizer is multi-threaded and will split the calculation over all the CPUs in your machine if that will speed the calculation up.
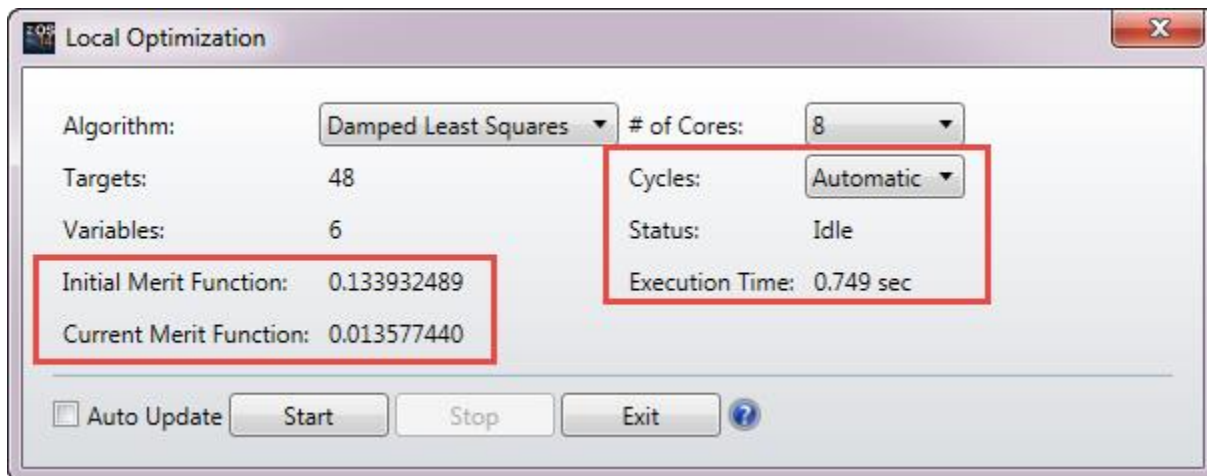
The merit function value quickly falls, and the Spot Diagram plot shows the improved performance (double-click it to make it update). The RMS spot radius is now 11μ on axis, and about 20μ at the 5° field point, compared to 143μ and 168μ prior to the optimization. That's a big improvement!
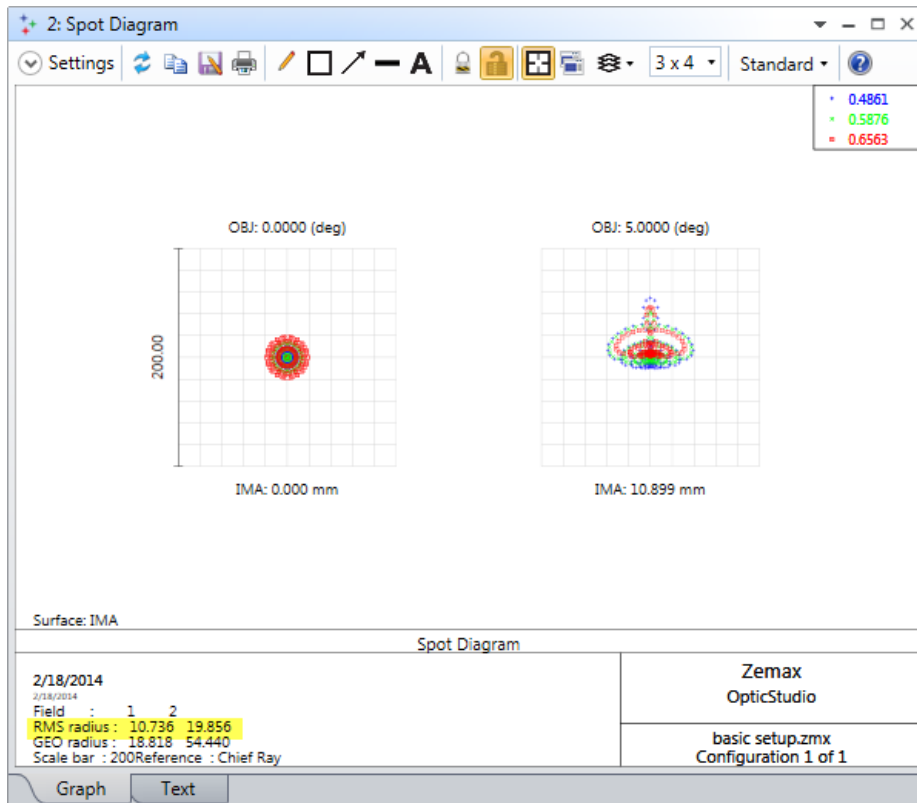
However, there is a clear problem, which can be seen in the Layout plot:
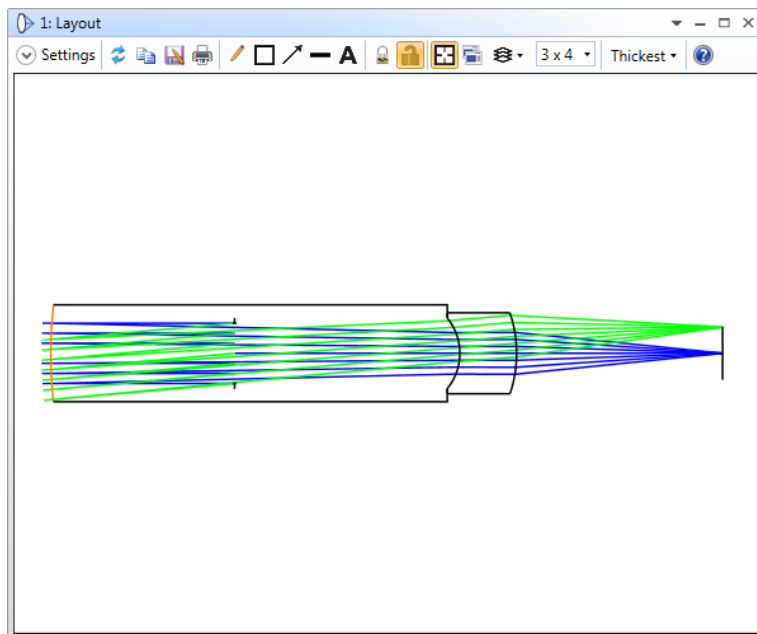
The lens is unfeasibly thick! We have told OpticStudio to minimize the RMS spot radius, but have given it no guidance about any constraints it must operate within. Press the F3 button, or click on the Undo icon:
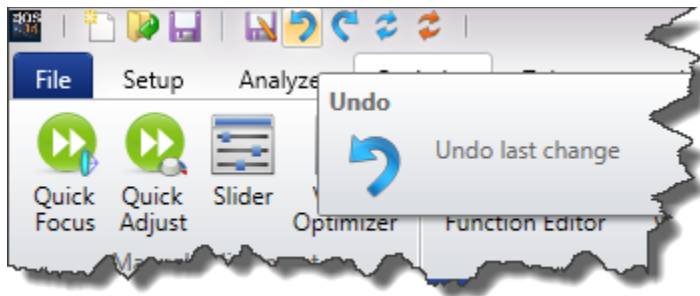
This will undo the optimization and restore the previous, un-optimized system. Click on the Optimization Wizard again (it is in the Merit Function Editor toolbar as well as the Optimize tab) and set up the following changes
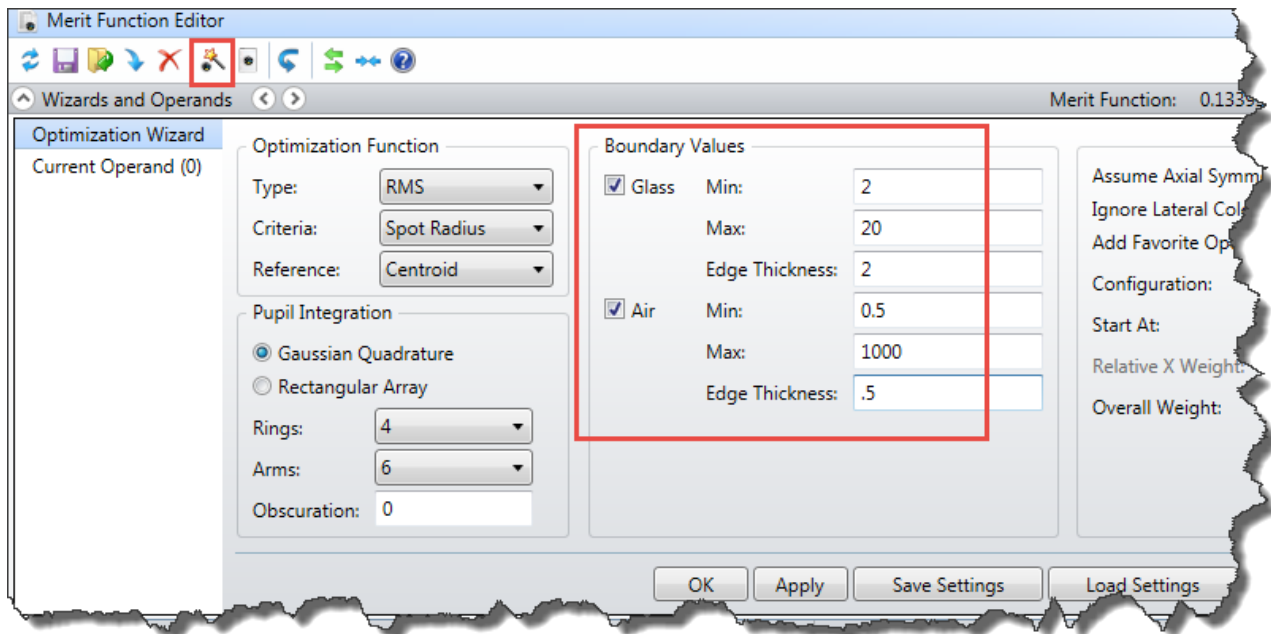
These settings require that the lens elements have a center thickness somewhere in the range between 2 and 20 mm, and that the lens edge-thickness be greater than 2 mm (this is a useful constraint to aid manufacturability). Any surfaces made of air must have thicknesses between 0.5 and 1000 mm, which is not necessary in this design, but in a multi-element design will prevent lens elements from hitting each other or being unreasonably far away from each other, and is therefore added here for completeness.

Click OK to enforce the changes. Press the Optimize! icon again, and we get a much better design:
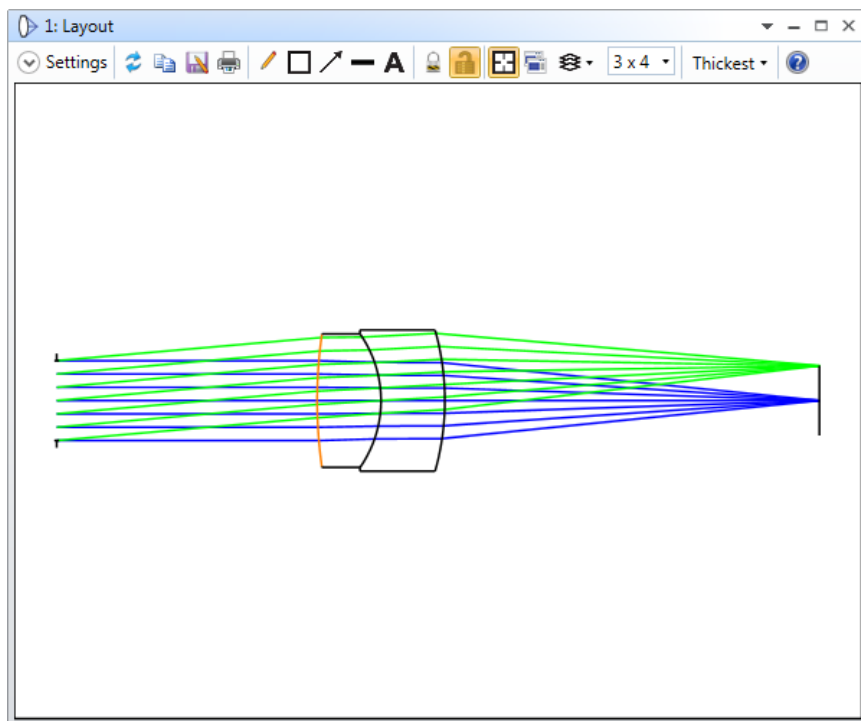
And the RMS spot radius is now 13.6μ on axis, and about 26.1μ at the 5° field point:
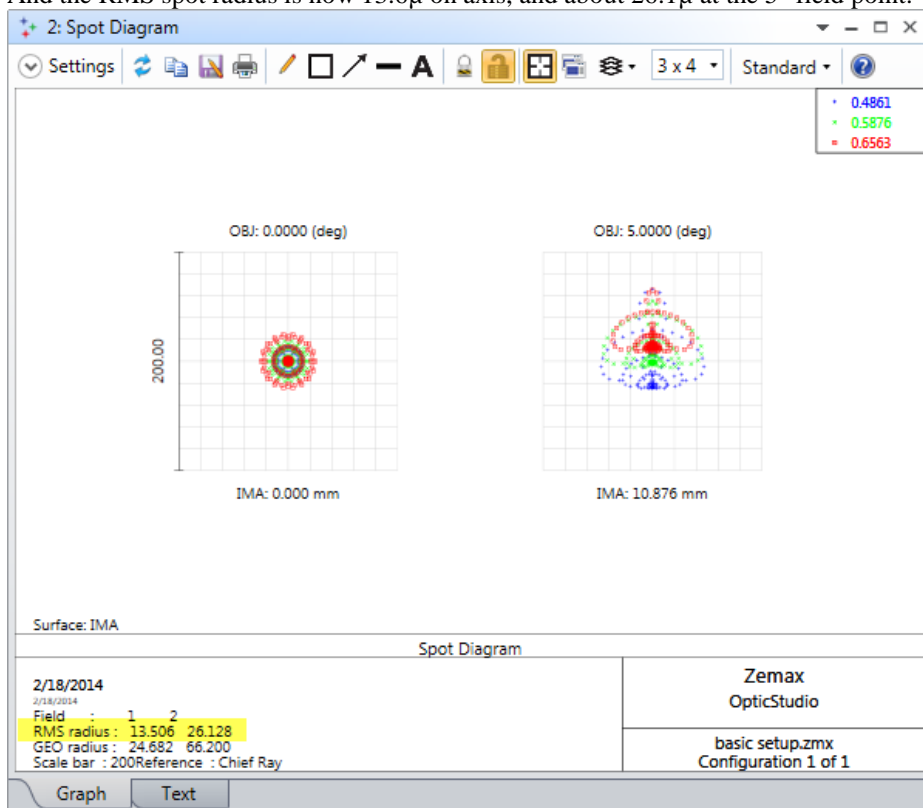
Note: The key point is that for successful optimization, the merit function should contain both the optical targets you want to achieve, plus constraints that will prevent OpticStudio from producing unwanted design shapes. Typical constraints include the thickness of elements, weight, maximum acceptable distortion, etc.

Are There Enough Field Points?

We optimized this lens using just two field points, at 0° and 5°. Although the RMS spot radius looks well controlled at these two points, how do we know that at some intermediate field point the performance of the lens does not degrade?

In the Analysis tab, click on RMS…RMS versus Field:
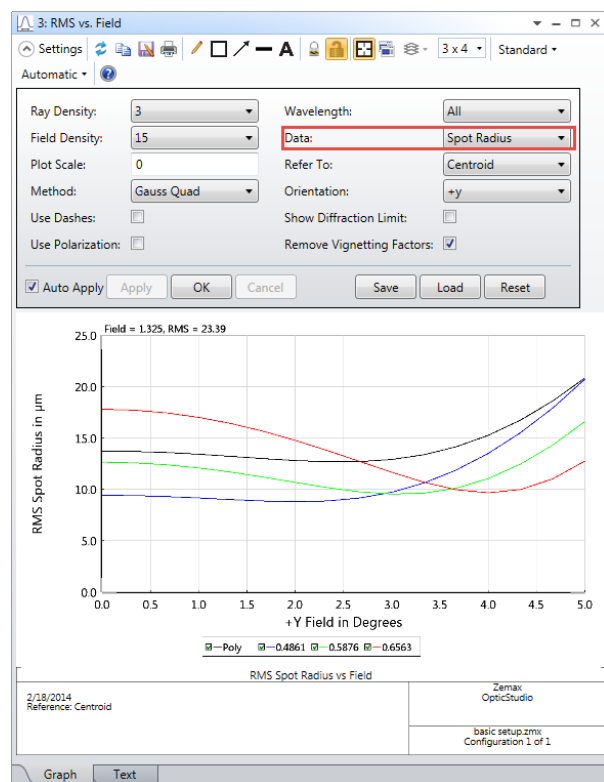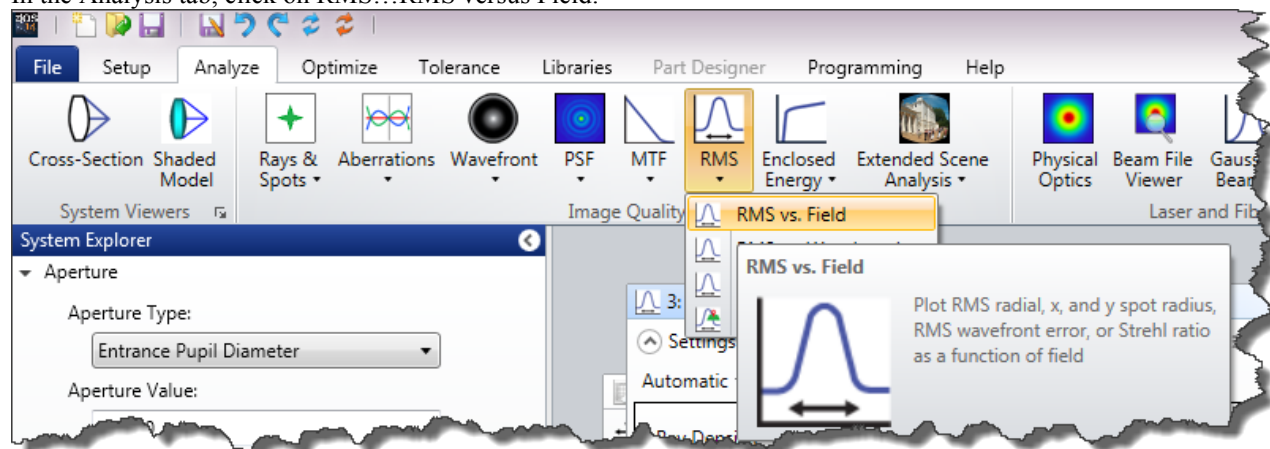
This plot shows how the RMS spot radius varies as a function of field, with field as a continuous variable. We are using 15 points across the 5° field, and plotting the RMS spot radius for each wavelength individually and as a polychromatic average. Note that the RMS spot never exceeds its value at the extreme fields of 0° and 5°. Therefore, two field points provide adequate control in this design. If the curve shows exceeds the value at the maximum or minimum field points, add more field points as required.

Note: If you change the number of field points, or the number of wavelengths, you must rebuild the merit function to include your changes into it.

A similar RMS vs. Wavelength plot allows you to check that you have adequate control with the defined number of wavelengths, as does Analyze… Aberrations…Chromatic Focal Shift and Analyzed…Aberrations…Lateral Color.

Another excellent way to look at the optical behavior over field and wavelength is to use Analyze…Extended Scene Analysis…Image Simulation. Configure it like so:
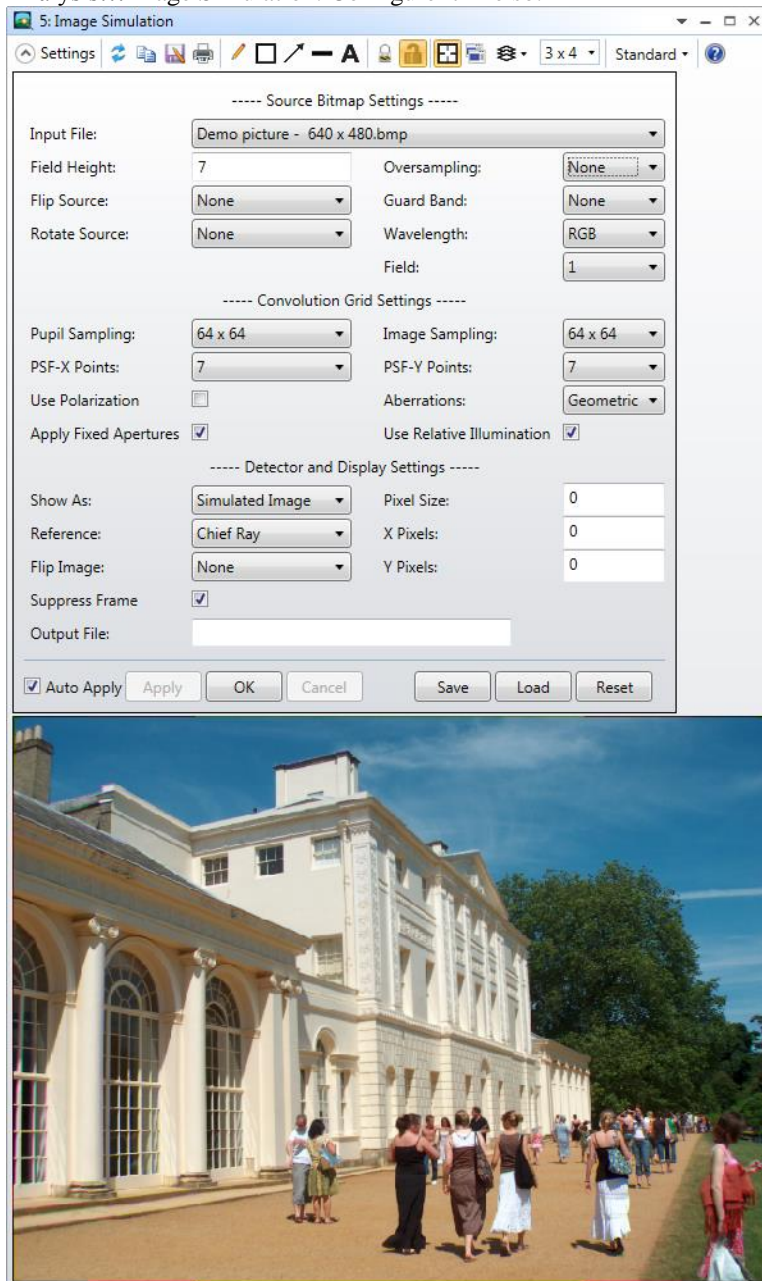
This will produce a simulation of what a real source scene, described by an input bitmap, will look like when imaged by the lens. This analysis is amazingly fast, taking literally only a few seconds to produce the image below. This feature is ideal for communicating real-world optical performance to non-optical specialists.

Glass Optimization

There is an important difference between optimizing glasses and other system parameters. Parameters like radii of curvature, thicknesses, etc. can be smoothly varied: a thickness of 10.0 mm can become 10.00001mm for example. However, glasses are only available with discrete properties: you cannot simply perturb a glass to get a slightly different refractive index! Instead, we use a method called Glass Substitution to swap out the glasses that the design currently uses for other glasses.

The first step is to define a template for the glasses OpticStudio is allowed to choose. Within the Optimize tab, choose the Glass Substitution Template from the Global Optimizers group:
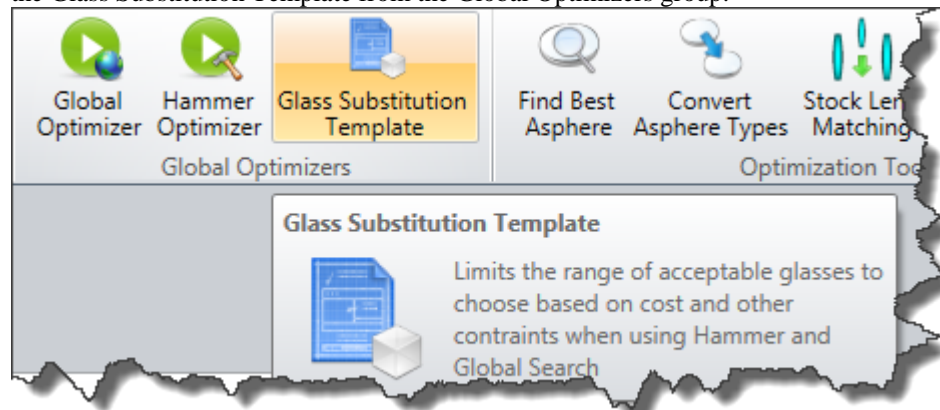
And configure it like so:
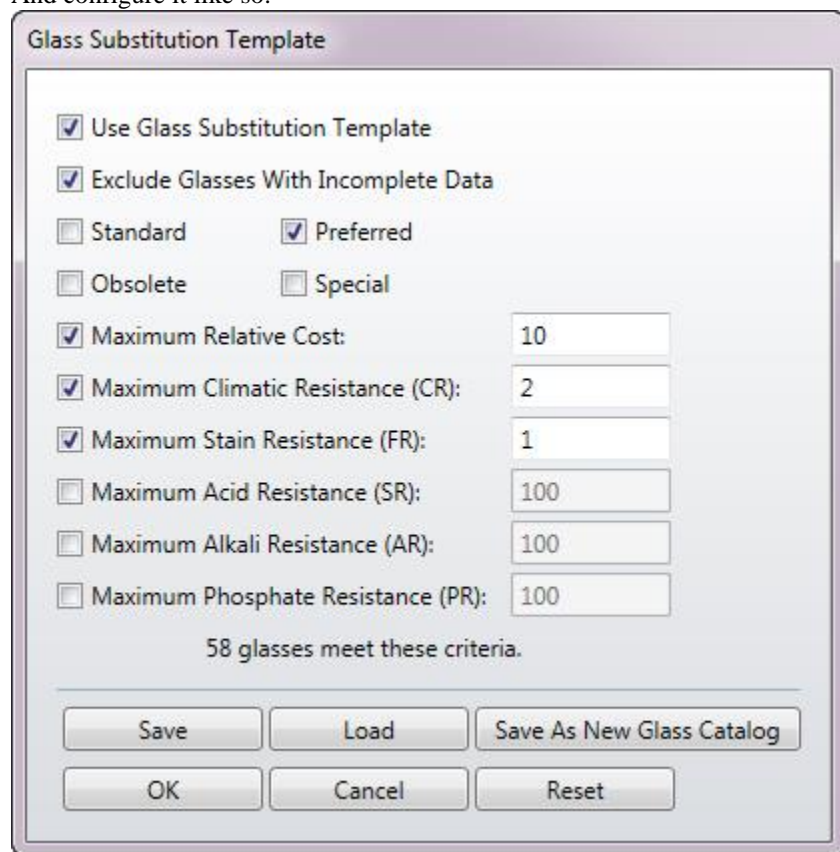
We are telling OpticStudio to use only Preferred optical glasses (a status flag that indicates the glass is easily available and does not have any unusual properties). In addition, each glass must cost no more than 10 times the price of N-BK7 (the relative cost), and must have a Climate Resistance factor of 2 or better and a Stain Resistance factor of 1 or better. There are a total of 58 glasses in the currently loaded catalog (by default the Schott glass catalog is loaded) that meet these criteria, and these are the only ones that will be selected for substitution.

Then click on the cell top the right of the material of surface number 2, which is currently N-BK7, and in the solve dialog box set a 'Substitute' solve:
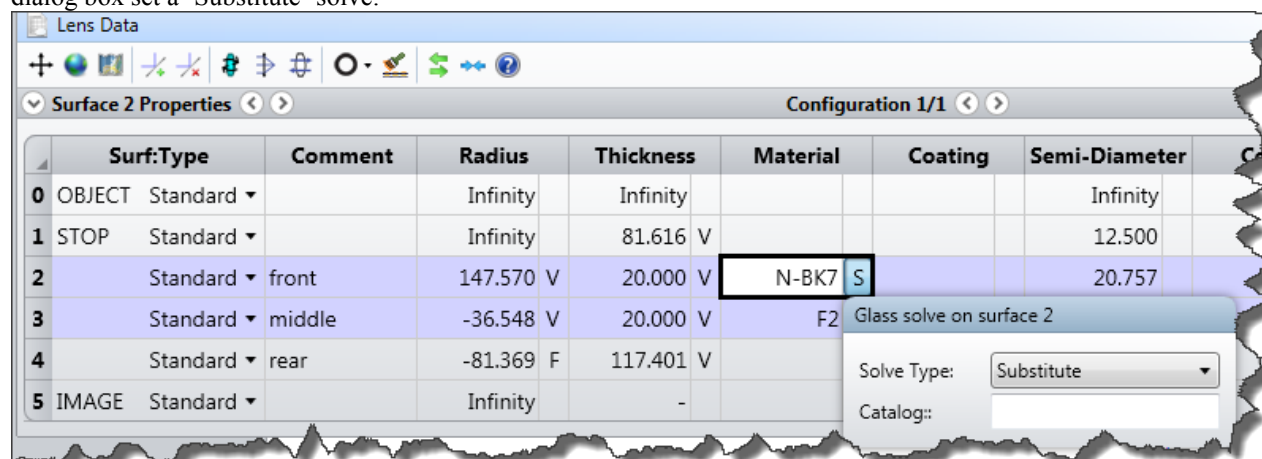
Repeat this for the F2 on surface 3. The Lens Data Editor should show an S status next to the glasses, to indicate that these glasses may be substituted:
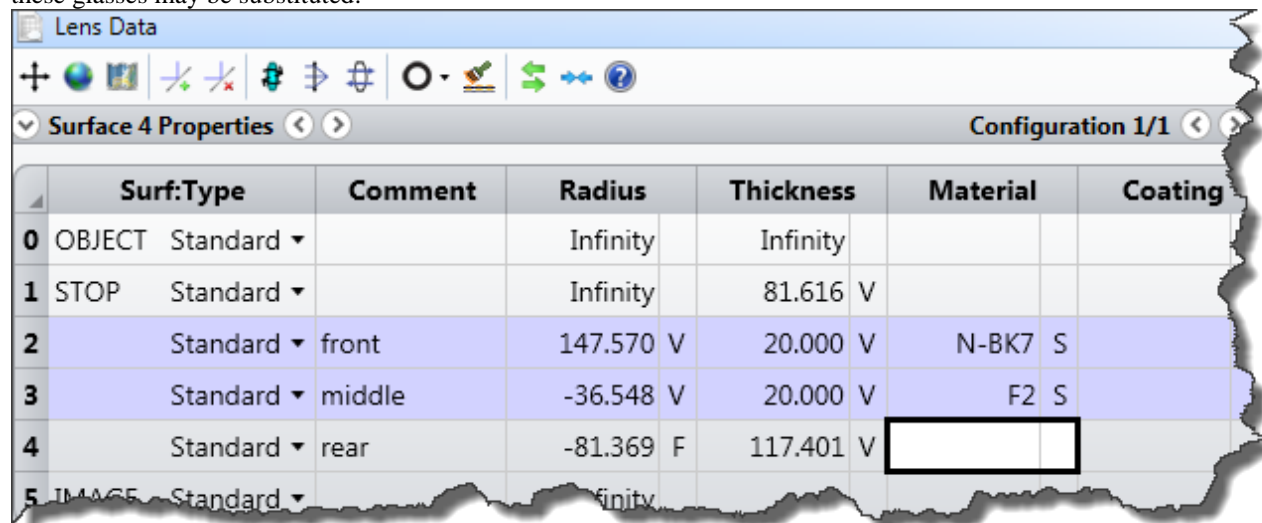
The glass substitution method is too complex for the local optimizer. Instead, use the Hammer optimizer in the Global Optimizers group and OpticStudio will quickly find the best glasses for this design:
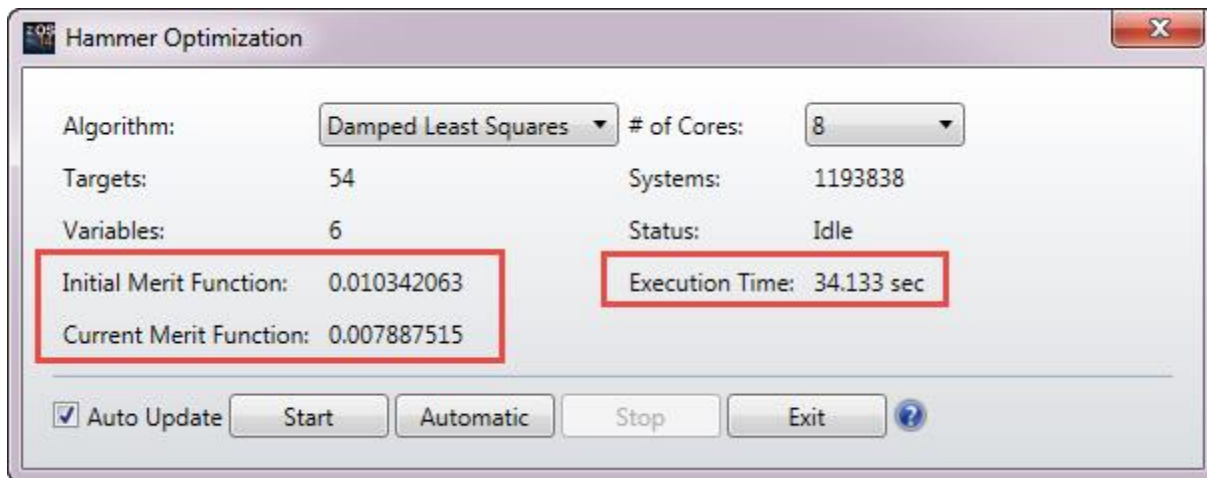
Check the glasses in the Glass Catalog to ensure they meet the specification (click on the glass, and go to Libraries…Materials Catalogs). The design now has an RMS spot radius below 19μ everywhere across the field of view:
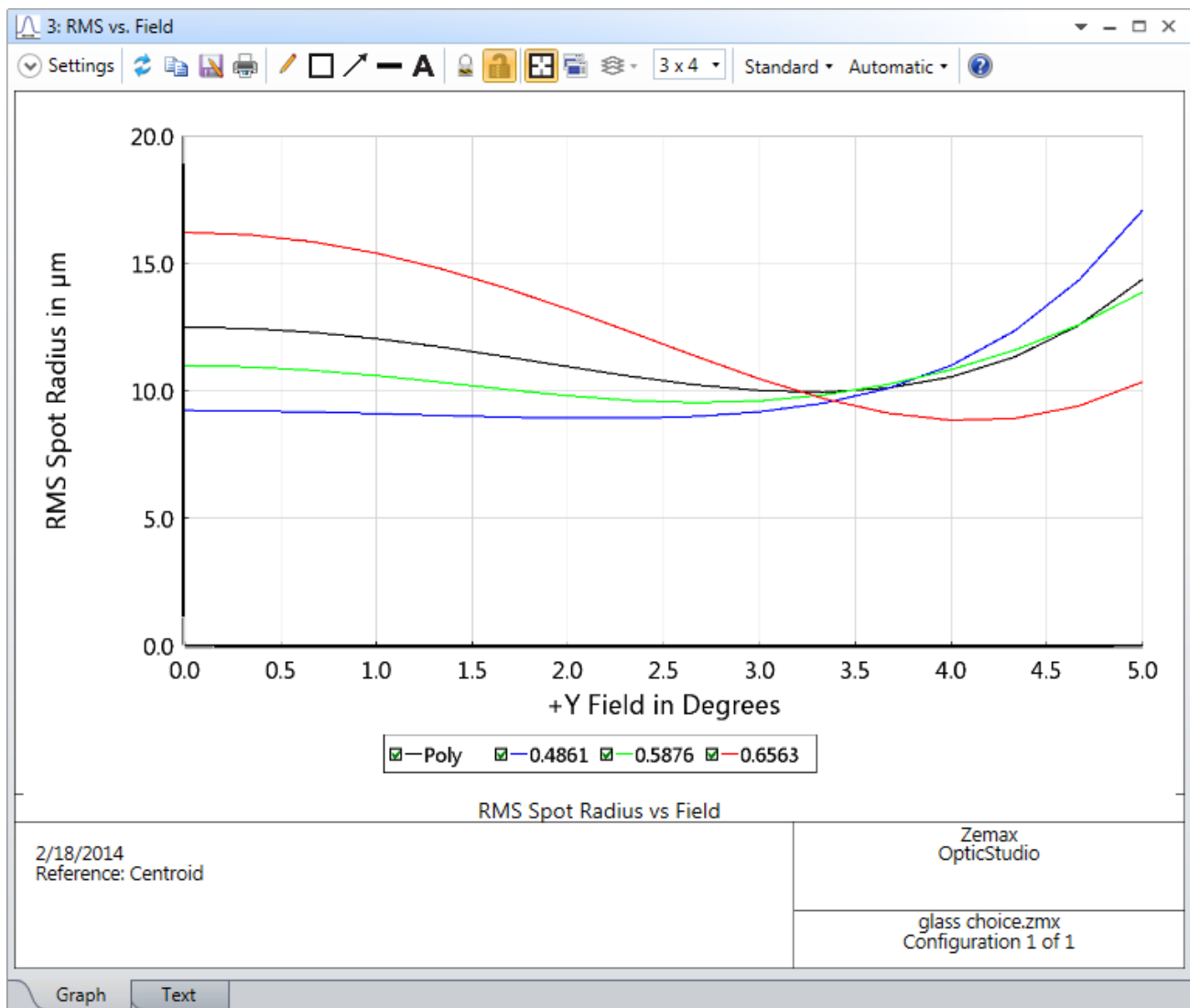
Tutorial 5.4: Tips and Tricks for Successful Optimization

The following tips and tricks are useful guidelines for how to make the best use of OpticStudio's wide and powerful optimization features.

Use Physically Significant Merit Functions

Before you start to design your lens, think about how it will be tested and used. Test methods fall into a number of broad categories:

- Imaging onto CCD arrays or (less common) photographic films. RMS Spot Radius is usually a good performance indicator in this case. If the final system is expected to have less than about 2 waves of aberration, use RMS Wavefront Error instead.
- If you will test your lens on an interferometer, optimize for RMS Wavefront error.
- If you will test your lens on an MTF measurement rig, use RMS wavefront error. MTF improves as RMS wavefront error approaches zero. If you need further improvement, use the various MTF* operands described in the Optimization chapter of the Technical Reference to target MTF performance at specific spatial frequencies.
- If you are designing an afocal system like a beam expander, switch the lens to Afocal Mode via the switch on the System Explorer's Aperture section. Use RMS Angular Radius if you expect more than about 2 waves of aberration in the final system, and RMS wavefront error if you expect less than 2 waves of aberration.

Don't Optimize Aberration Coefficients Directly

It is tempting to attempt to 'control' the lens under optimization by targeting Seidel aberrations like SPHA, COMA, etc. directly in the merit function, and then using fifth-order aberrations for better control (see the macro fifthord.zpl, or the optimization macro ZPL03.zpl for example). While this is perfectly possible to do in OpticStudio, we do not recommend it, for the following reasons:

- Aberrations are difficult to compute in tilted and decentered systems, or systems with components like aspheric surfaces, diffractive components or GRINs.
- The Gaussian Quadrature (GQ) method OpticStudio uses for RMS Spot Radius and RMS Default Merit Functions are exact to a specified order of wavefront aberration. If you use n rings in the default merit function, you then have control of all wavefront aberrations up to order $r^{(2n-1)}$. In the doublet lens we designed, we used 4 rings and therefore could control all aberrations up to r^7, which is a higher order than fifth order aberrations can achieve. See G. W. Forbes, "Optical system assessment for design: numerical ray tracing in the Gaussian pupil", J. Opt. Soc. Am. A, Vol. 5, No. 11, p1943 (1988) for a very readable and full account of this useful technique.
- We are usually interested in optimizing for real-world performance metrics like spot size, wavefront error, MTF, etc. Fifty years ago aberration theory was a useful computational shortcut, but 21st century computers and multi-threaded software like OpticStudio are massively faster than the tools available then. Optimizing directly for the desired performance is more practical than optimizing for some intermediate function that we hope will then go on to give us the desired performance. Optimize for what you want to test the built system for!
- Note for example that in the optimization of the doublet we did not need to target chromatic effects like axial or lateral color directly: the default RMS spot radius merit function provided this automatically.
- The exception to this is distortion, because distortion affects only the location of the image, not its quality. Operands like DIMX, DISG, etc. can be used to control distortion.

Use the Optimization Wizard

We recommend that you use the Optimization Wizard as the foundation of your merit function construction. This generates a 'Default Merit Function' which is the bedrock of your optimization function. Ultimately imaging systems are characterized by RMS spot radius or wavefront error, and afocal systems by RMS angular radius or wavefront error. The Optimization Wizard also automates the construction of the most common opto-mechanical constraints designers require, such as lens edge and center thickness constraints.

Additional goals and constraints can be easily added by inserting your own operands above the Wizard's operands in the editor. OpticStudio writes out the dummy operand DMFS (Default Merit Function Start) to indicate where the default merit function starts, and you should not hand-edit the operands below this line. Just click on the DMFS operand and press the insert key to insert new lines above the default merit function.

Use Hammer Often

The Hammer optimizer is used to improve a lens that has already been optimized by the local optimizer. Get in the habit of leaving the Hammer to run on an optimization problem when you are working on something else. The Hammer can be left to run overnight, or over weekends or even weeks if necessary.

Use Adequate Boundary Conditions

You should always add boundary constraints to your merit function, as well as optical targets. This yields two important benefits:

- OpticStudio will produce designs you can build, and that meet your non-optical goals! For example you should always constrain edge and center thicknesses of lenses to be reasonable, and you can add constraints on length, weight, etc. as required by your application.
- Good boundary constraints speed up the global optimizers because OpticStudio does not look in regions of parameter space where the boundaries are violated. This can speed up Global Search in particular by orders of magnitude.

Use the System Check Utility

The System Check utility, on the Setup tab, is an invaluable aid to ensuring there are no accidental errors in the system setup. Although not every possible fault can be found by such a utility, anything it does report should be checked, and anything classed as an 'Error' must be rectified.

Tutorial 6: Non-Sequential Ray Tracing (Professional and Premium only)

Note: Even if you intend to use only the non-sequential mode of OpticStudio, you should work through the previous tutorials before starting this one. Sequential and non-sequential modes share many common user interface concepts and methods which are described in those chapters and are not repeated here.

Non-sequential ray-tracing is a powerful and general technology for tracing rays in systems where there are multiple optical paths. Typical uses include:

- Illumination systems, especially those with multiple or complex optical sources
- Systems such as interferometers, in which light that has travelled through several different optical systems must be coherently recombined
- Opto-mechanical stray light analysis in otherwise sequential optical systems
- LCD backlighting
- Bio-optical systems, particularly those based on scattering from tissue or fluorescent scattering

Non-sequential ray tracing assumes that there is no pre-defined path for any ray. A ray is launched and hits whatever object is in its path, and it may then reflect, refract, diffract, scatter, split into child rays, etc. It is a far more general technology than sequential ray-tracing, and is therefore somewhat slower in terms of ray-tracing speed but more general in its scope.

Sequential designs can be converted easily to non-sequential mode by using Convert to NSC Group on the File tab.

Tutorial 6.1: A Simple Example

Click on File…New to start a new OpticStudio design. Then click on Setup…Non-Sequential UI Mode. A new editor, the Non-Sequential Component Editor, (NSCE) will appear. The Analyze and Tolerancing tabs are not the same as in sequential mode.

The NSCE is very similar to the Lens Data editor or the merit function editor in look and feel, and if you know how to use these editors the NSCE holds no surprises. The NSCE consists of a toolbar, Property Inspector and spreadsheet region, just like the other Editors

However, in non-sequential mode we deal with 'components' or 'objects' rather than 'surfaces'. Objects are full 3D volumes, not a collection of individual surfaces.

There are three basic types of object:

- Source objects, from which rays are launched into the non-sequential system
- Geometry objects, which define the optical components (lenses, prisms, mirrors, CAD objects, etc.) that the rays reflect, refract, scatter or diffract from
- Detector objects, which detect rays and give quantitative data of optical performance like irradiance, radiant intensity etc.

Open the Properties Inspector for Object 1 in the NSCE:
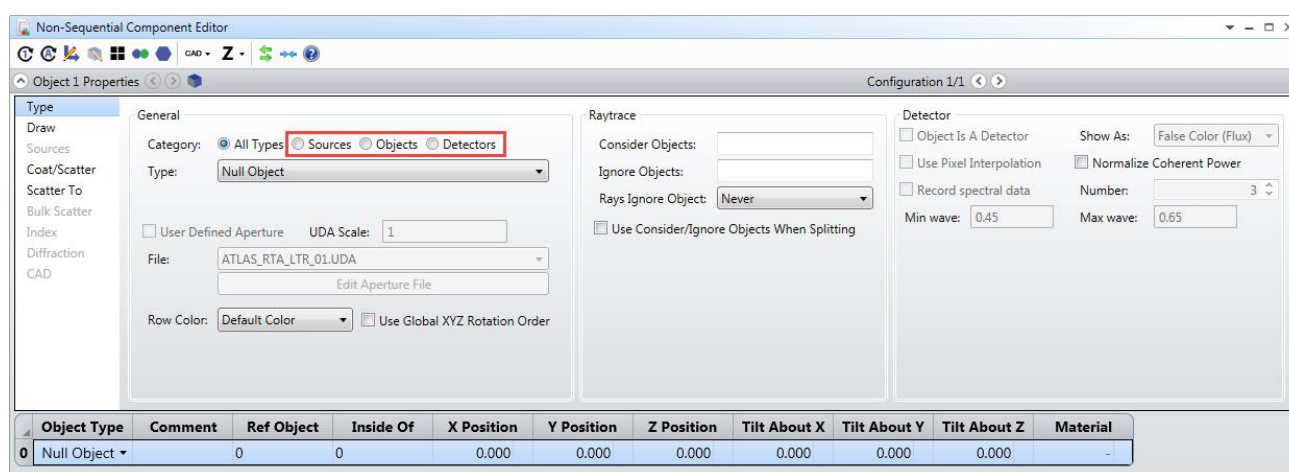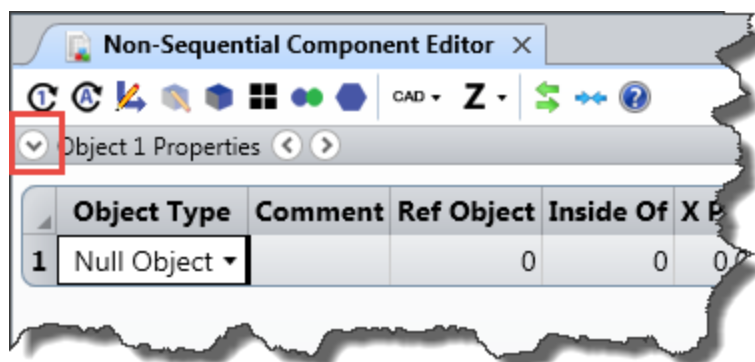
Select 'Objects' to restrict the listing to geometry objects, and select 'Standard Lens' from the list:
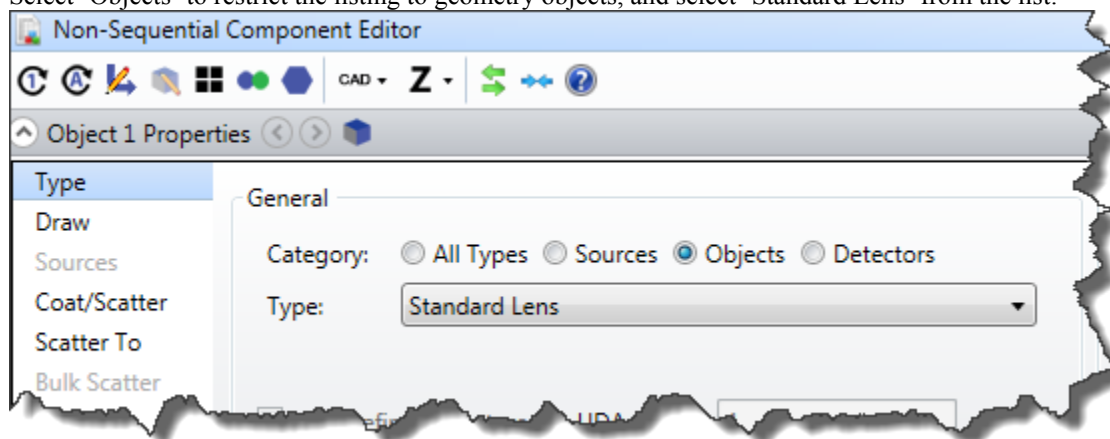
And close the Property Inspector. Note you can also just type directly into the Object Type cell in the editor, or use the drop-down list:
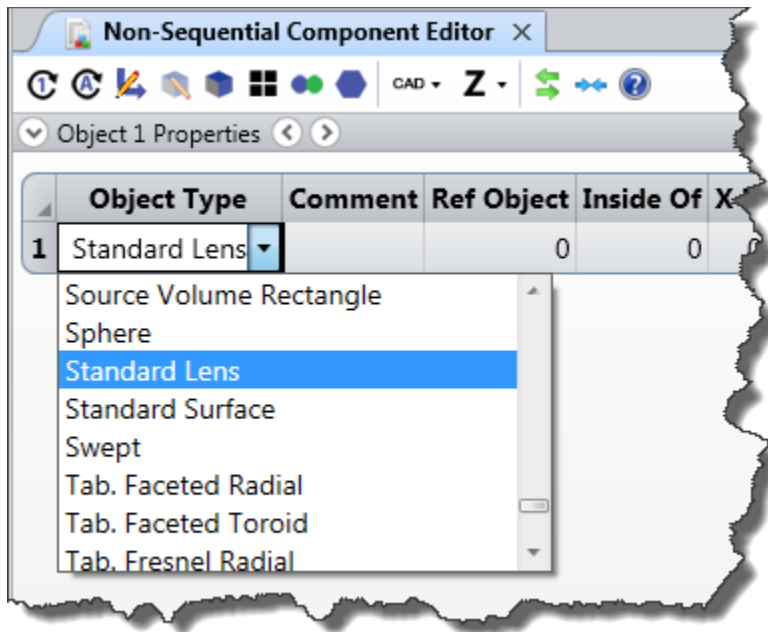
Open a Shaded Model plot to see the object and use a Y|Z cutting plane to make the layout more meaningful.
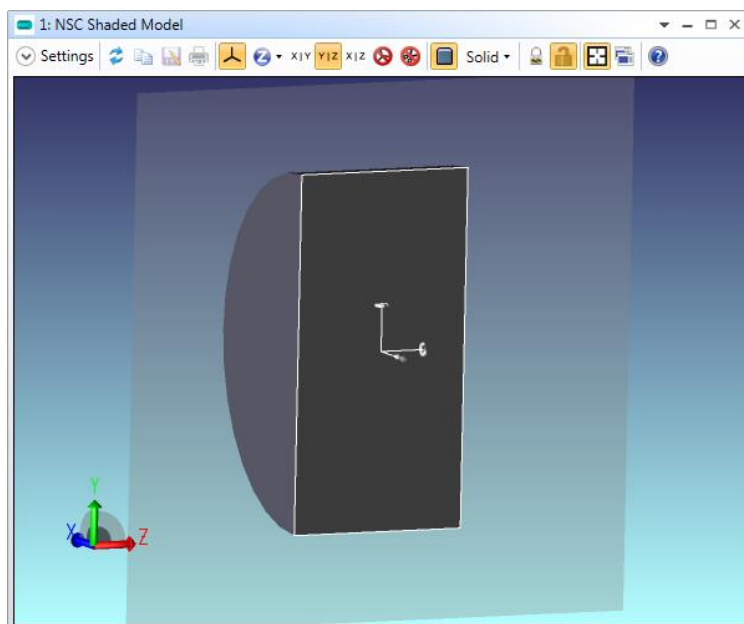
Note that you have a single 'Standard Lens' object that defines the entire object, rather than two surfaces and a thickness. Looking at the editor, you can position this object at any (x, y, z) location, and tilt it about x, y, z. You can then enter the glass type the object is made from, and its defining parameter data. Enter the following data:

```
All positions and tilts: 0.0
Material:                 N-BK7
Radius 1:                 5.0
Conic 1:                  0
Clear 1, Edge 1:          both 1.0
Thickness:                1.0
```

```
Radius 2:                        2.0
Conic 2:                         0
Clear 2:                         0.8
Edge 2:                          1.0
```

You should see this:
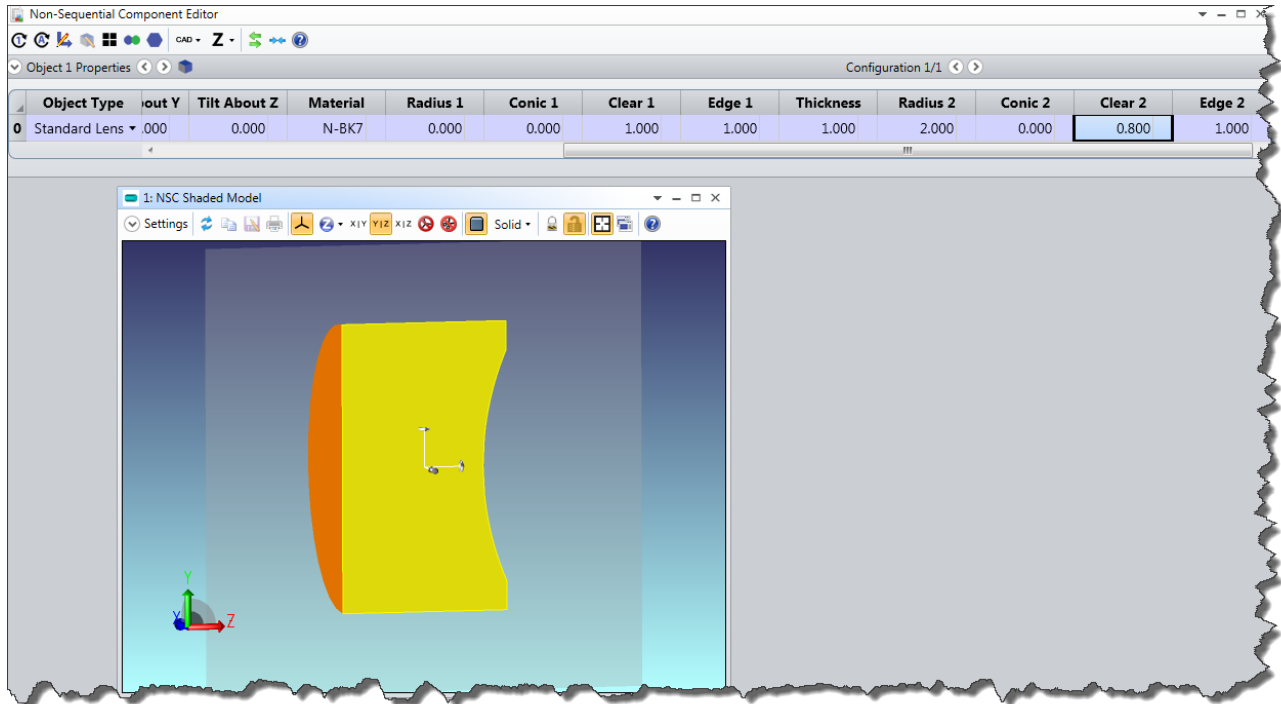
This is a fully parametric lens, modeled as a solid object and not a collection of surfaces.

Next, click on the lens object in the editor again and press the Insert button to create a new 'Null Object' and make it a 'Source Ray' object.
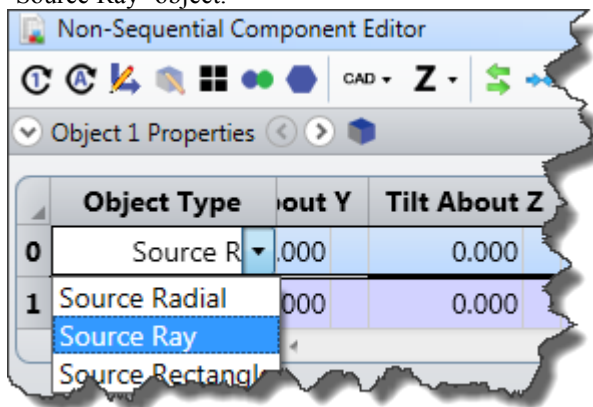
Enter the following data:

```
All parameters are 0.0 except:
Y position:                      0.5
Z position:                      −1.0
Tilt About X:                    15.0
Layout Rays:                     1
```

You will then see this ray-trace like so:
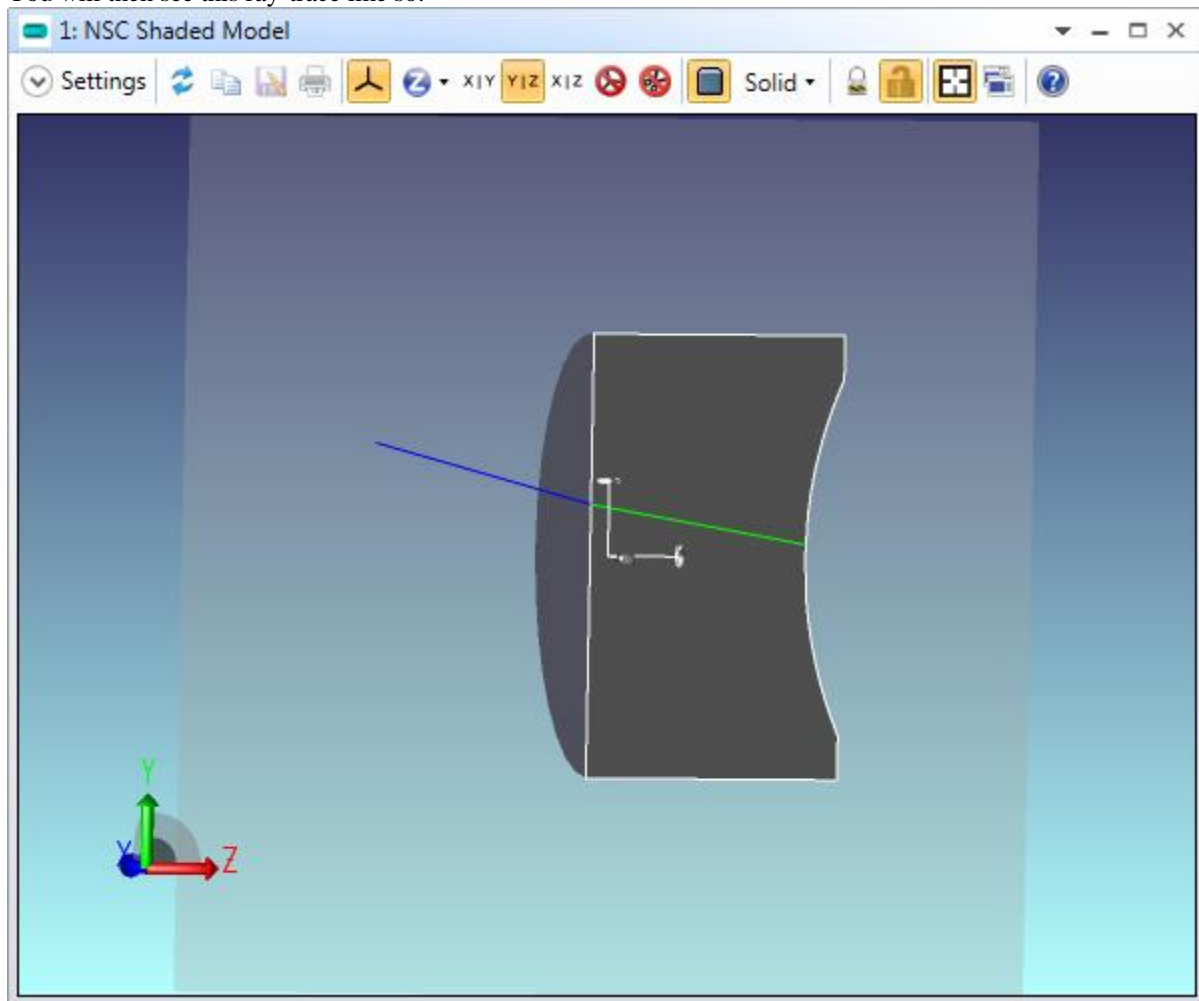
You may have to adjust the slicing plane manipulator slightly to see the whole ray trace. The ray is traced from the source to the front face of the lens, and then onto the second face of the lens. As there is no further object for the ray to hit, OpticStudio draws it for a short distance and then stops tracing it.

Now click on the row number of the Source Ray object to select the whole row, right mouse click and select 'Copy Object'.
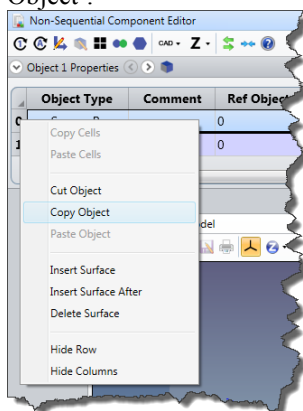
Click on the row again so it is no longer highlighted and press <Cntl>V or right-mouse click again and select Paste Object to paste the object back into the editor as a new object. You should now have two identical source objects. Modify the parameters of one of them as follows:

```
Z position:      2.0
Tilt About X:    -15.0
Tilt About Y:    180.0
```

Do not change any other parameters. Update the layout as you make each change so you can see what is happening. You should see two sources, one on either side of the lens.
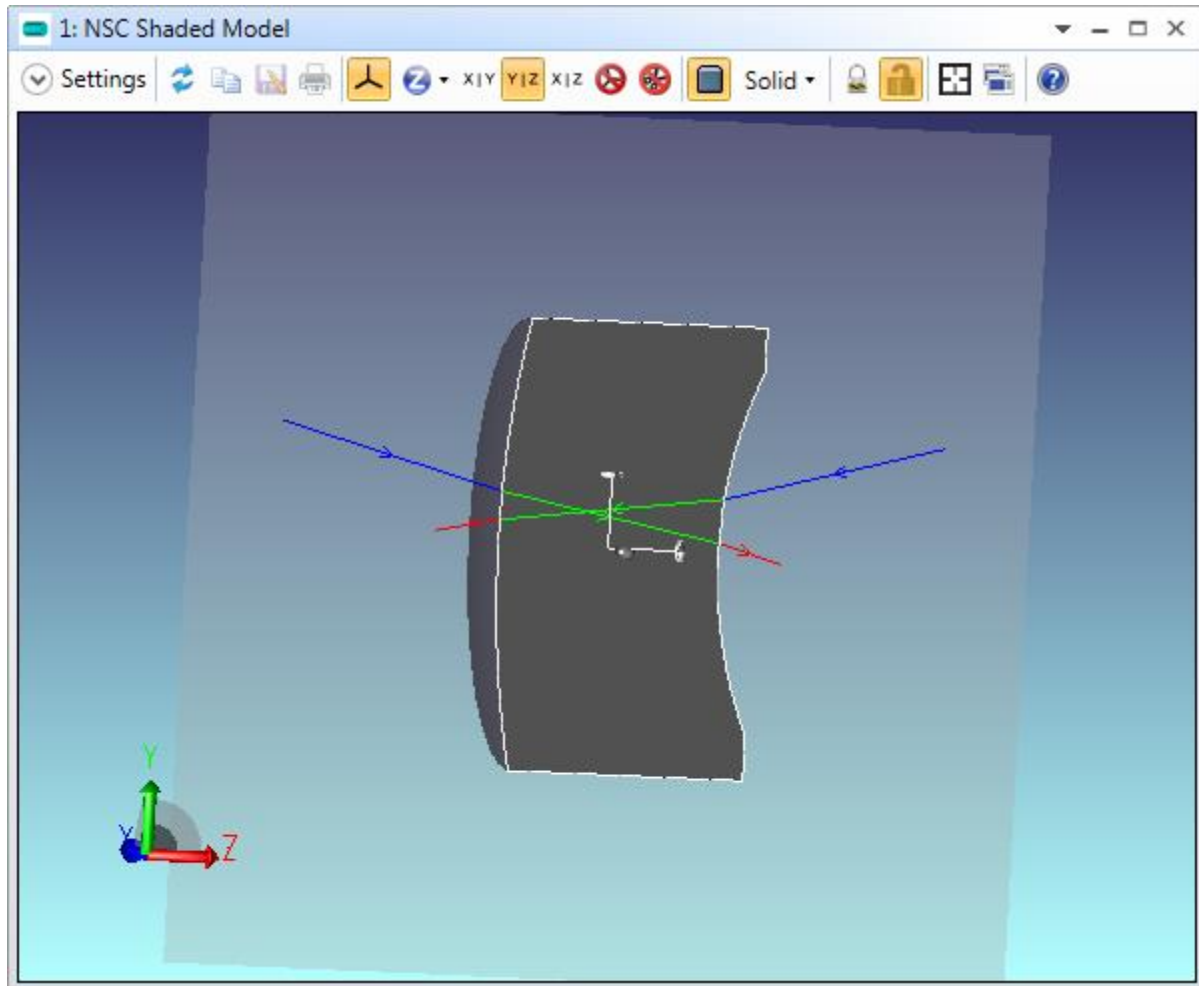
Now, on the Settings of the Layout plot, select the options to 'Split Rays' and 'Fletch rays':
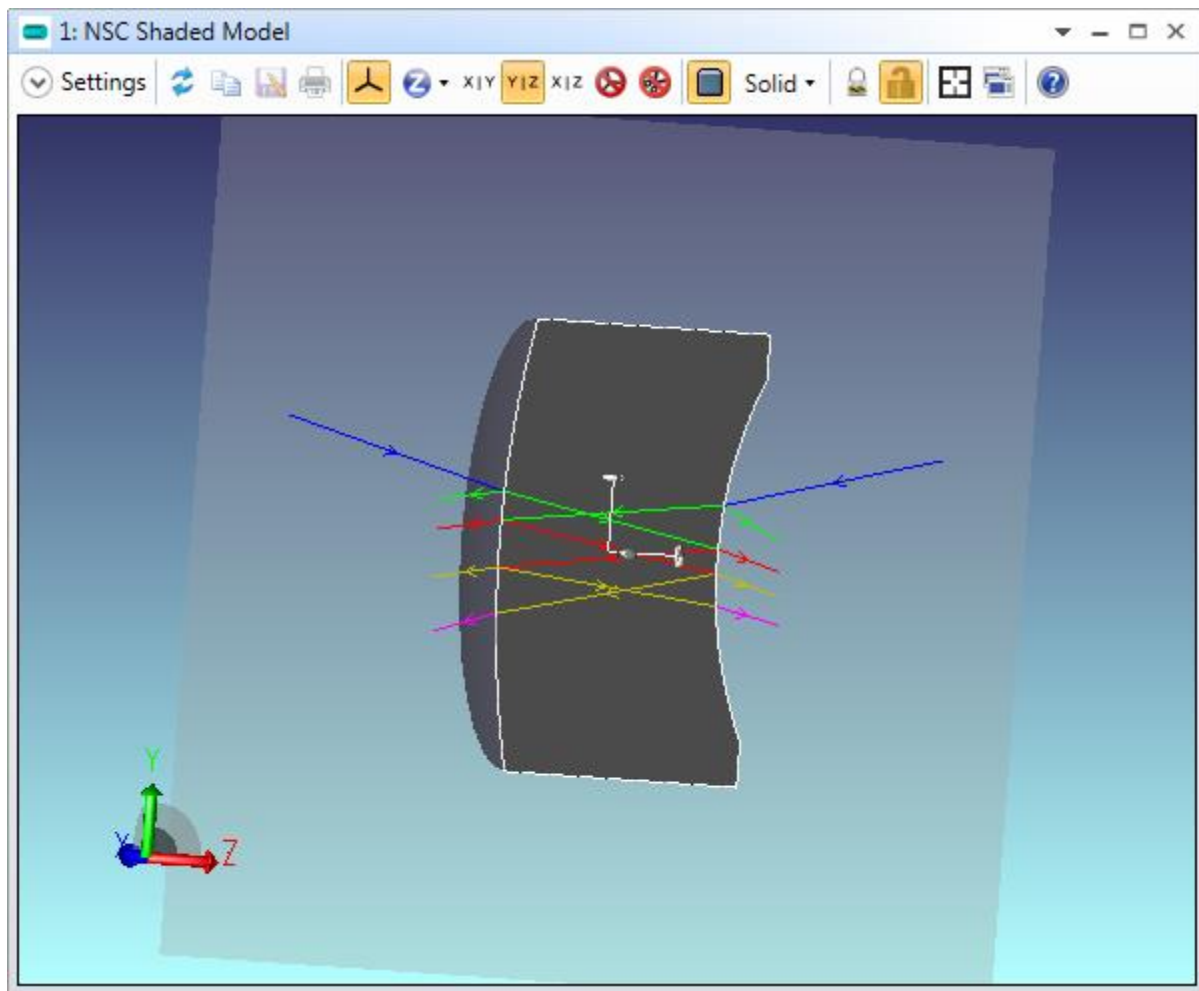
This simple example shows the key benefits of non-sequential ray tracing:

- You do not have to tell rays where to go. Rays are launched and then interact with whatever objects are in their path.
- When a ray hits a refractive object, part of its energy is reflected and part is transmit-ted. OpticStudio can produce 'child' rays that take the reflected energy, and these child rays then interact with whatever is in their path, and in turn produce children of their own, which can have children of their own, etc.
- As well as being partially reflected and refracted, rays can also scatter at the surface of an object or inside its volume (called bulk scattering to distinguish it from surface scattering).
- Sources, objects and detectors are placed in a global coordinate system, and can be positioned and tilted independently of each other. In addition, if it is required, objects can be positioned relative to other objects, which we will discuss later.

Because rays can be split into transmitted, reflected and scattered components, as each ray splits it contains less and less energy. We need to put some limits on the ray-tracing to prevent OpticStudio from tracing rays with insignificant amounts of energy. This is defined in the System Explorer, in the Non-Sequential grouping:
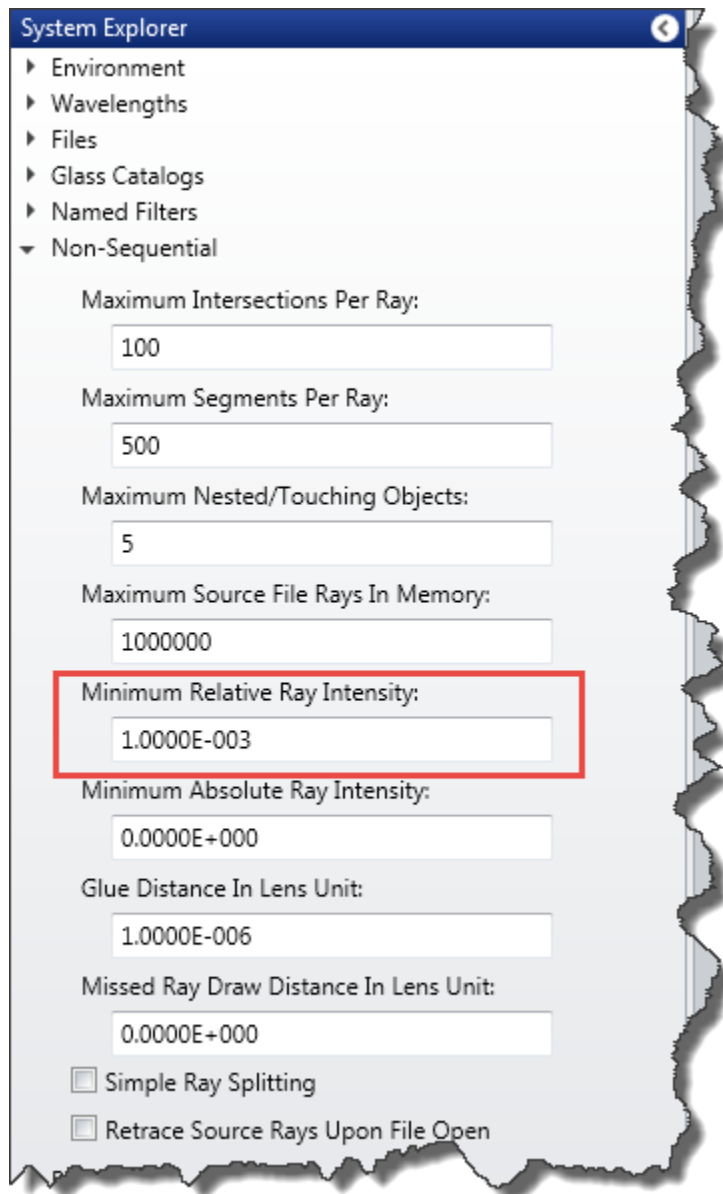
Try varying this parameter and observe how it affects the number of child rays produced. Set it to 10^{-2} and note you get fewer rays; at 10^{-12} you will get many more.

Tutorial 6.2: Object Positioning & Definition

The Non-Sequential Component Editor provides an easy way to define the non-sequential optical system, and the inter-relationships between components. Open the sample file in the Samples\Non-sequential\Miscellaneous folder called "Digital Projector Flys Eye Homogenizer.zmx". The system contains an elliptical source volume that approximates the shape of the fireball in an arc lamp inside a parabolic mirror. The output light enters a homogenizing optical system that consists of two fly's eye lenslet arrays and a field lens. The homogenizer is manufactured as a complete sub-system which is then placed into the optical beam produced by the source and parabolic mirror.
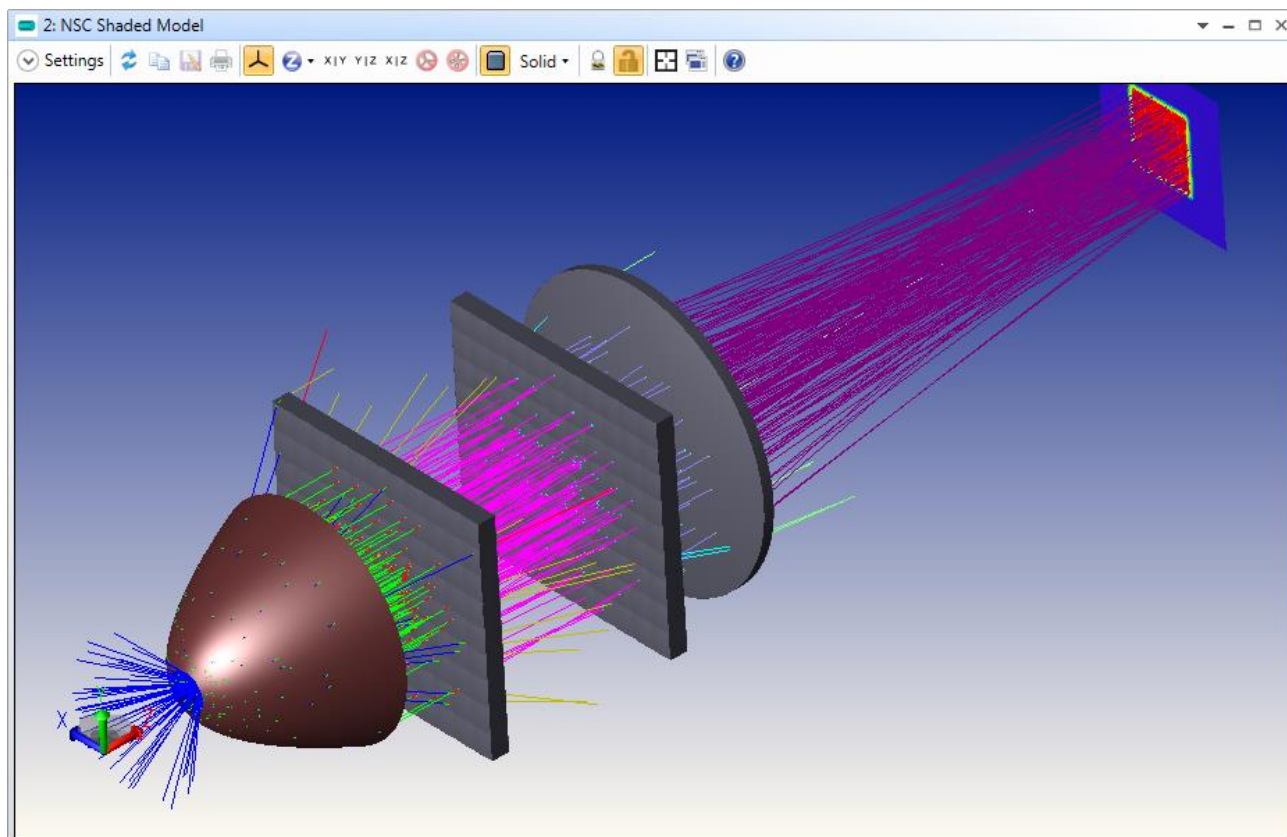
Object Positioning

Note how the objects are referenced:

Every object has a number, shown in the left-most column of the editor, and a 'Reference Object'. Ref Object 0 is the global coordinate reference point of the whole 3D space, and objects 1, 2 and 3 are positioned relative to this coordinate system. Objects 4, 5, 6, and 7 are positioned relative to object 3, and they therefore are positioned like a sub-assembly:

try moving object 3, and notice that objects 4-7 automatically move as well. So the position of object 3 defines the position of the homogenizer assembly.

Note: Any object can be positioned relative to any prior-defined object, which can be positioned relative to any other object defined prior to it, etc.

Imagine we now want to move the homogenizer about some arbitrary point. Click anywhere on object 1, and press the Insert key, so that you now have a 'Null Object' as object 1, and all other object numbers have automatically incremented. The Lenslet Array object that serves as the reference for the homogenizer assembly is therefore now object 4.

Null objects have no optical properties, and so they are useful for defining reference and pivot points, for example. Position the null object at y = 40, z = 70, and then double-click on the object type to show the object properties tab:
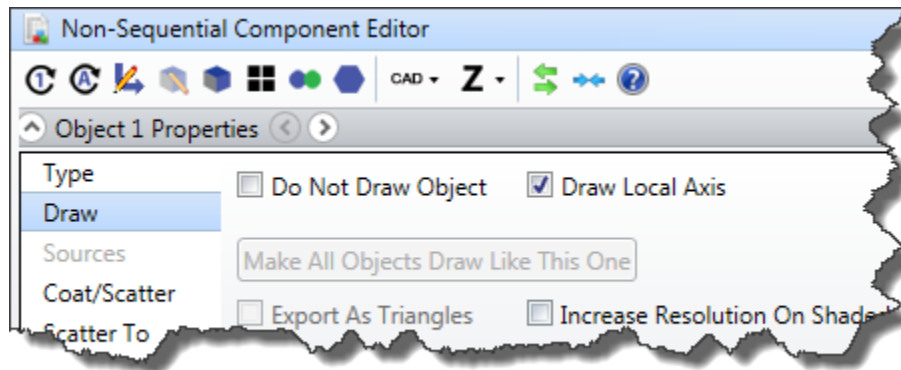
On the Draw tab, check the 'Draw Local Axis' control. On the NSC 3D Layout, the Null object's local axes are now drawn (note that local axes are never drawn in the Shaded Model).

3D Layout

Digital Projector Fly's Eye Integrator
2/25/2014

Zemax
OpticStudio

Digital_projector_flys_eye_homogenizer.ZMX
Configuration 1 of 1
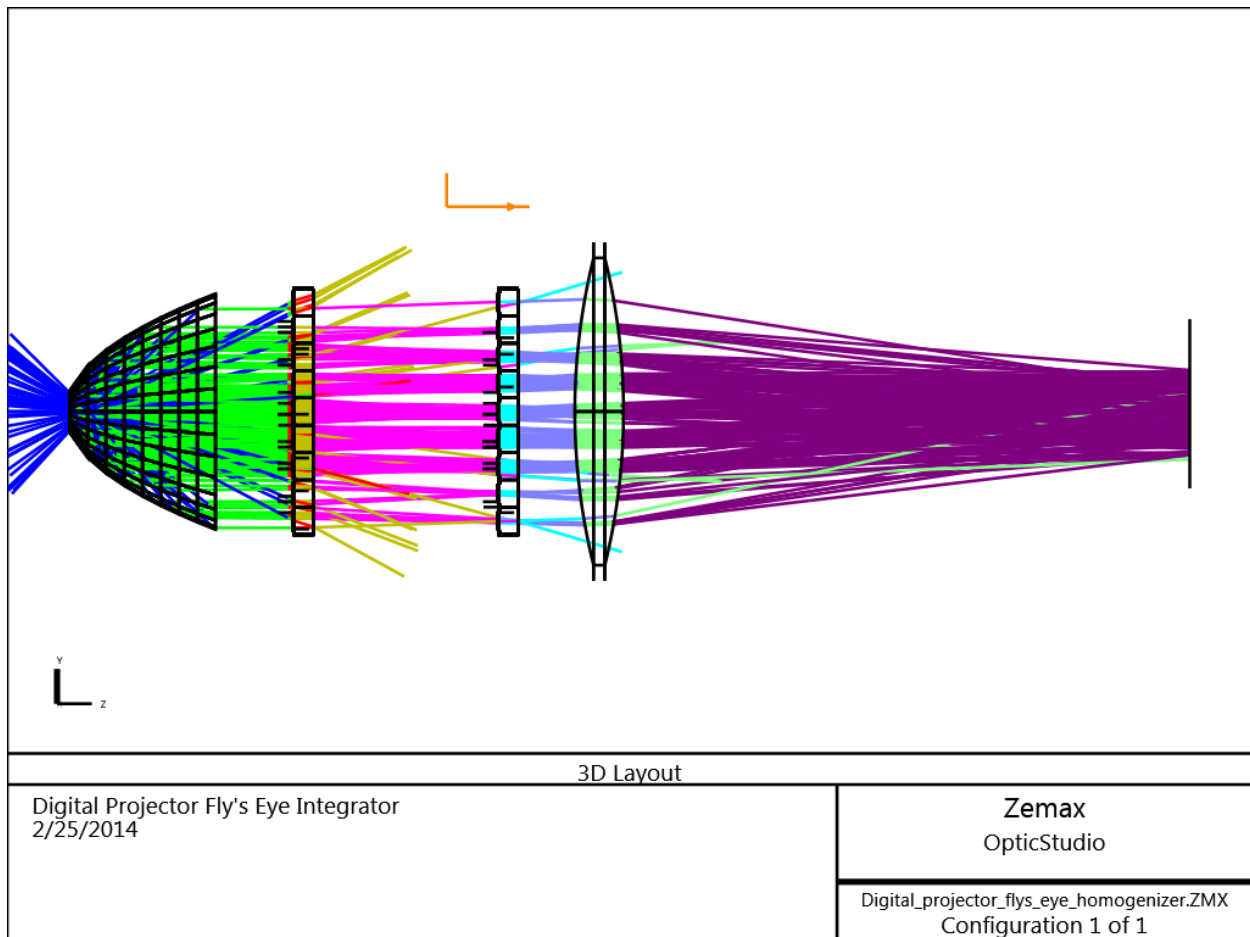
On the Non-Sequential Component Editor's toolbar, choose the Modify Reference Object tool:
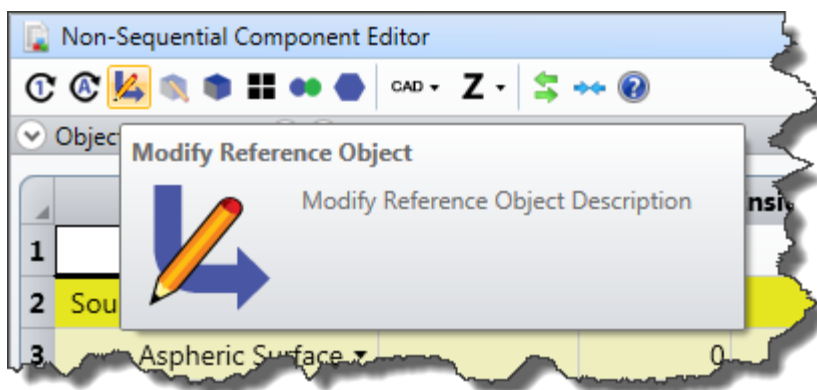
And set it like so:
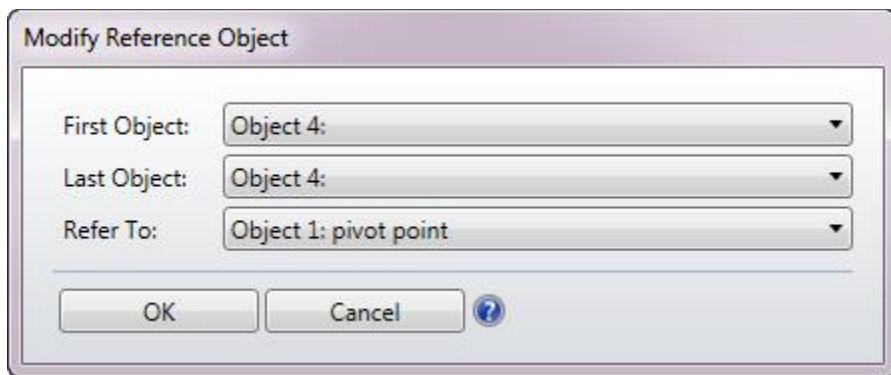
OpticStudio will now modify object 4's properties so that it is positioned relative to object 1, while retaining its absolute position and orientation in global coordinates. In other words, object 4 has not moved, but its position is now defined relative to a different object. The subsequent objects are still positioned relative to object 4:

If you now apply a Tilt About X of 10 degrees to object 1, you will see that the whole homogenizer assembly pivots about object 1, but the lamp assembly remains in place:

3D Layout

Digital Projector Fly's Eye Integrator
2/25/2014

Zemax
OpticStudio

Digital_projector_flys_eye_homogenizer.ZMX
Configuration 1 of 1
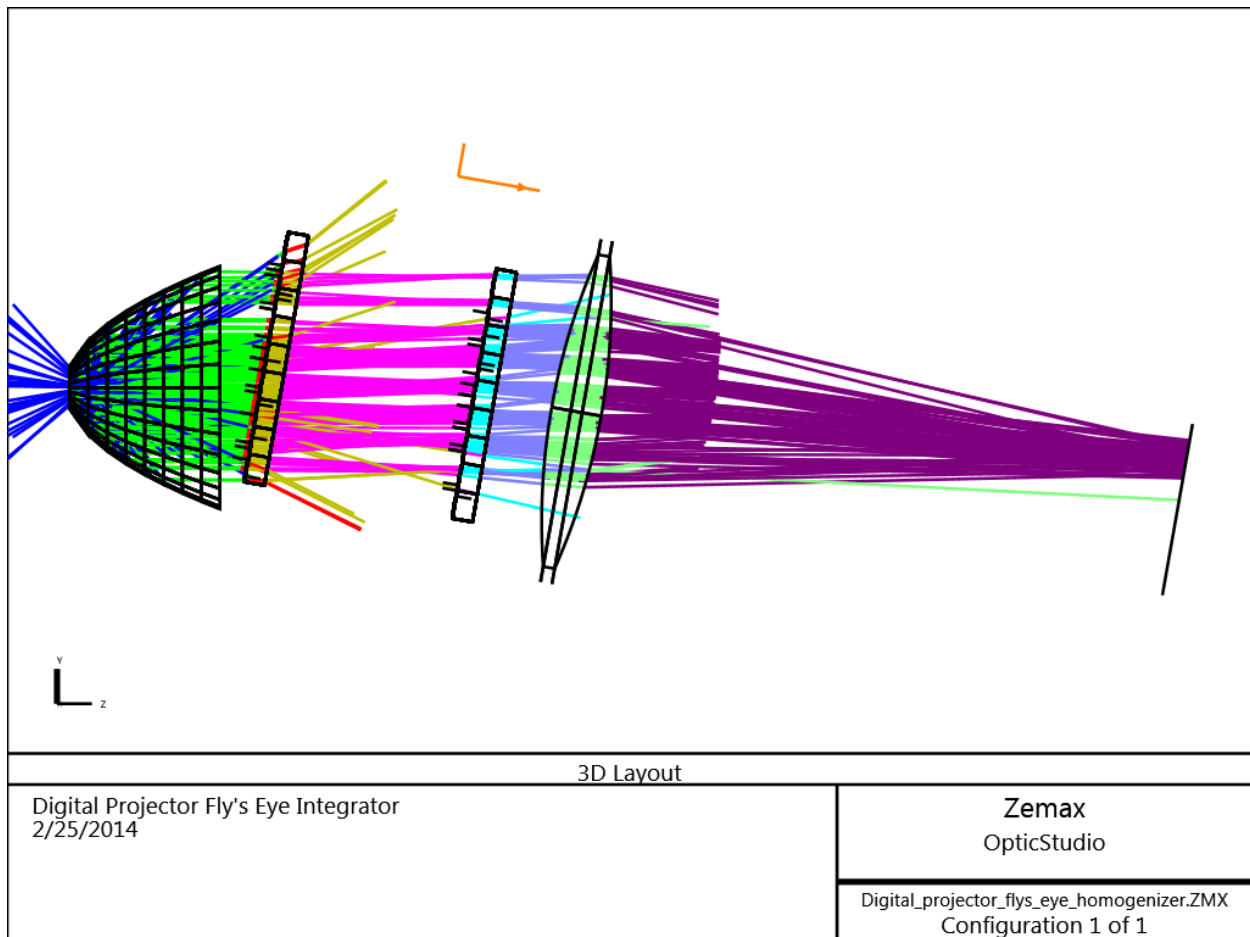
Object Parameters

As well as being positioned, objects are given their defining parameters in the NSC Editor. For example, the Lenslet 2 object is defined by parameters like x and y halfwidths, thickness, radii of curvature and conic constant, and numbers of lenslets in x and y.

	Object Type	Material	X Half-Width	Y Half-Width	Thickness	Radius 1	Conic 1	Radius 2	Conic 2	Number In X	Number In Y	Par
1	Null Object ▾	-										
2	Source Volume Ellipse ▾	-	250	1000000	1.000E+004	0.250	0.250	0.500				
3	Aspheric Surface ▾	MIRROR	9.000	-1.000	23.000		0					
4	Lenslet Array 2 ▾	B270	2.318	2.667	4.000	21.000	0.000	0.000	0.000	11	9	
5	Detector Rectangle ▾		22.000	22.000	100	2	0	0	0.000	0	0	
6	Lenslet Array 2 ▾	B270	2.318 P	2.667 P	4.000	20.500	0.000	0.000	0.000	11 P	9 P	
7	Standard Lens ▾	BK7	125.000	0.000	30.000	-125.000	0.000	30.000	30.000			
8	Detector Rectangle ▾	ABSORB	15.000	15.000	100	2	0	0	0.000	0	0	
9	Null Object ▾	-										

There is a great advantage to this parametric approach. Parametric objects require relatively little memory, are fast to ray-trace, and can be changed easily. They are also optimizable, just like sequential surfaces.

Further, interrelationships between objects can be easily defined via pickup solves. Note that the second lenslet object uses pickups on several parameters to lock itself to the first lenslet object (Note the letter P on some of the defining properties of this lens, which denotes a pickup solve). This is a great advantage during optimization, as changes to one parameter can automatically flow through the whole system.

Tutorial 6.3: Combining Sequential and Non-Sequential Ray-Tracing

Most imaging systems are well described by the orderly sequential approach used in the Lens Data Editor. However, there are cases where an otherwise sequential system has some region in which there is a need for non-sequential ray-tracing. Prism assemblies are a classic example, in which different parts of the beam interact with different faces of the prism(s) in a different order to other parts of the beam.

For example in Samples\Sequential\Afocal\Binocular_System.ZMX:
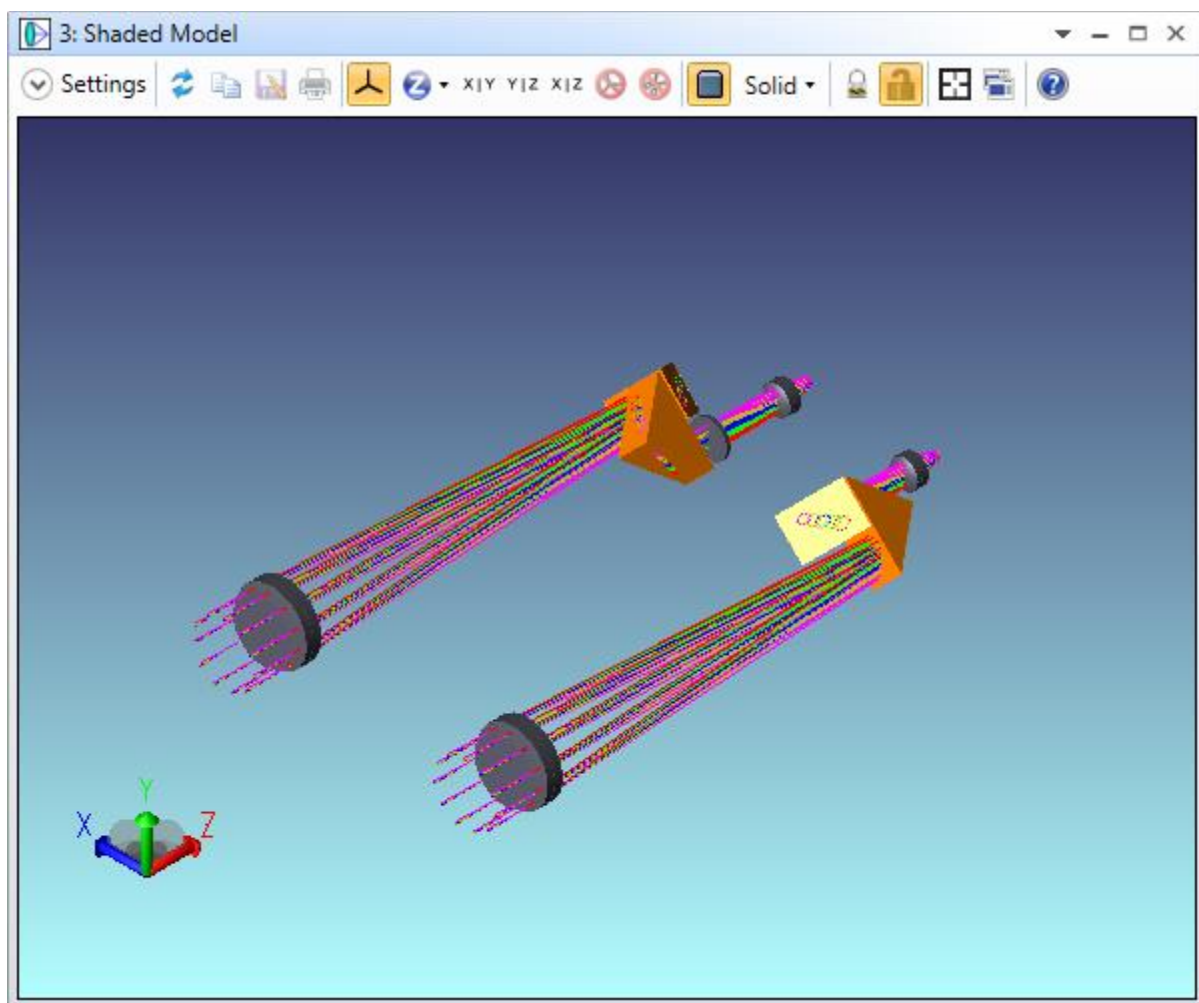
We have a classic sequential doublet objective and a Kellner eyepiece, just like any other sequential system. Note that a special sequential surface type called a **Non-Sequential Component** is used between the lens and eyepiece. This acts like the 'entry port' into the non-sequential world defined in the non-sequential component editor.

The parameters on the Non-Sequential Component surface in the Lens Data Editor define the location of the 'exit port', which is how rays come back to the sequential ray tracer. This is referred to as 'hybrid' or 'mixed' Sequential\Non-sequential ray tracing.

When a sequential ray hits the Non-Sequential Component surface in the Lens Data Editor, it is passed to the non-sequential ray-tracer and it interacts with whatever objects are defined in the NSC Editor and are in the ray's path. When the ray hits the region defined by the exit port in the Lens Data Editor, it is transferred back to the sequential ray-tracer and interacts with the subsequent sequential surfaces.

Note: Sequential rays cannot split inside the non-sequential group.

A sequential system can contain any number of Non-Sequential Component surfaces. The objects 'inside' each Non-Sequential Component group are independent of each other. You can easily switch between non-sequential component groups by clicking the Surface in the NSCE menu bar. This surface corresponds to the surface number of the NSC group in the Lens Data Editor, and is only shown if there is more than one NSC group inside the Lens data Editor.
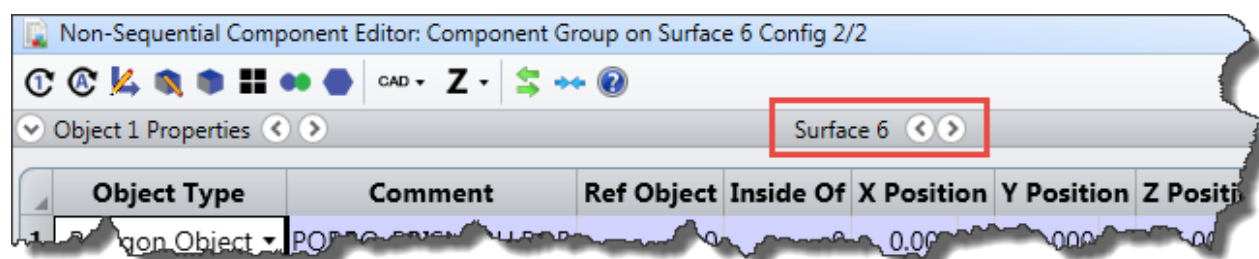
If a ray does not hit the exit port, it is terminated and it does not return to the sequential ray-trace.

Note: If the marginal and chief rays are not traceable through NS groups, OpticStudio cannot compute important sequential parameters like pupil positions and f/#.

Tutorial 6.4: Tracing Rays and Getting Data

Ray tracing in hybrid non-sequential mode works exactly as it does in sequential mode, except that there is no paraxial ray-tracing inside an NS group. In pure non-sequential we use source objects to launch rays and detector objects to get quantitative information.

Source Objects

Source objects are objects that launch rays into the optical system with the appropriate spatial and angular distributions to represent the radiance of the real sources in your system.

Source objects fall into two categories:
- Parametric sources, like the Source Diode or Source Filament, in which the source radiance is computed from some equation, and you enter the parameters for this equation via the editor.
- Measured sources, like the Source IESNA, Source EULUMDAT, and Source File. Note that IESNA and EULUMDAT data files contain only far-field (angular) data and model the source as a spatial point. The .DAT and .SDF formats, used by the Source file object, contain both spatial and angular ray data, and so define the full radiance of the source. Data in these formats are provided free by many LED and lamp manufacturers, and can also be exported by third party programs like Radiant Zemax' ProSource and Opsira's Luca Raymaker. The .DAT and .SDF formats are documented in the Users' manual in both ASCII and binary formats. The difference

between the two formats is that the .SDF format contains spectral (wavelength) data, whereas the .DAT format does not.

Source wavelengths are defined in the System Explorer Wavelength group, just like sequential systems, although other definitions are also available, and will be discussed later in the section on Colorimetry. Source Units (Watts or Lumens) are chosen under the System Explorer Units group.

Sources are positioned in global coordinates in exactly the same way as any other object. All sources use parameters 1-5 of the Non-Sequential Component Editor to define some basic information about the source. These are:

- # Layout Rays: Defines how many random rays to launch from the source when creating layout plots. This is typically a small number, say less than 100, and is used only for drawing purposes.
- # Analysis Rays: Defines how many random rays to launch from the source when per-forming analysis. This is typically a much larger number and may be millions or even billions of rays.
- Power (units): Power is the total power over the defined range of the source. The power units are specified by the system source units in the System Explorer.
- Wavenumber: The wavenumber to use when tracing random rays. Zero means poly-chromatic, which chooses ray wavelengths randomly with the weighting defined on the wavelength data editor.
- Color #: The pen color to use when drawing rays from this source. If zero, the default color will be chosen. The RGB values of each pen are defined under Project Preferences.

Parametric sources will then use further parameters to define their radiance.

Source objects are not made of any material. Rays, once launched, have no further interaction with source objects. Rays are normally launched into air, but can be launched inside some other refractive index if desired. Define a geometry object with the correct shape and refractive index first and then locate the source object inside it, and then use the "Inside Of" parameter to tell OpticStudio to launch the rays inside that object's refractive index.

Note: Source objects must use the Inside of parameter when the source is placed inside a geometry object, otherwise incorrect ray-tracing will result.

Detector Objects

Rays are detected by Detector objects. Almost any kind of geometry object can be used as a detector also, but the dedicated Detector objects are designed for the task of displaying spatial and angular data and provide the controls users need to represent data the way it is measured experimentally.

The most common type of detector object is the Detector Rectangle. This is a two-dimensional array of pixels, similar to a CCD array. It is most commonly used with the material ABSORB, so that rays terminate upon being detected, but it can also be set as a MIRROR (with coating, if required, to be discussed later), or its material may be left blank to indicate air. Note that when the material is left blank, rays are not perturbed in any way by being detected. This can be useful, but care must be taken as the detector may appear to not conserve energy if a ray interacts with it multiple times without losing any energy!

Tracing Analysis Rays

Open the sample file Samples\Non-sequential\Sources\Simple LXHL-BD01 LED model.ZMX. This file contains just two objects, a Source Radial set with data taken from the LumiLeds LXHL-BD01 LED datasheet, and a Detector Rectangle set to 100 by 100 pixels.

Note that the source object has a total power of 27 Lumens, uses 30 layout rays for drawing purposes, and uses 1 million Analysis rays. In the Analyze tab, click the Ray Trace icon:
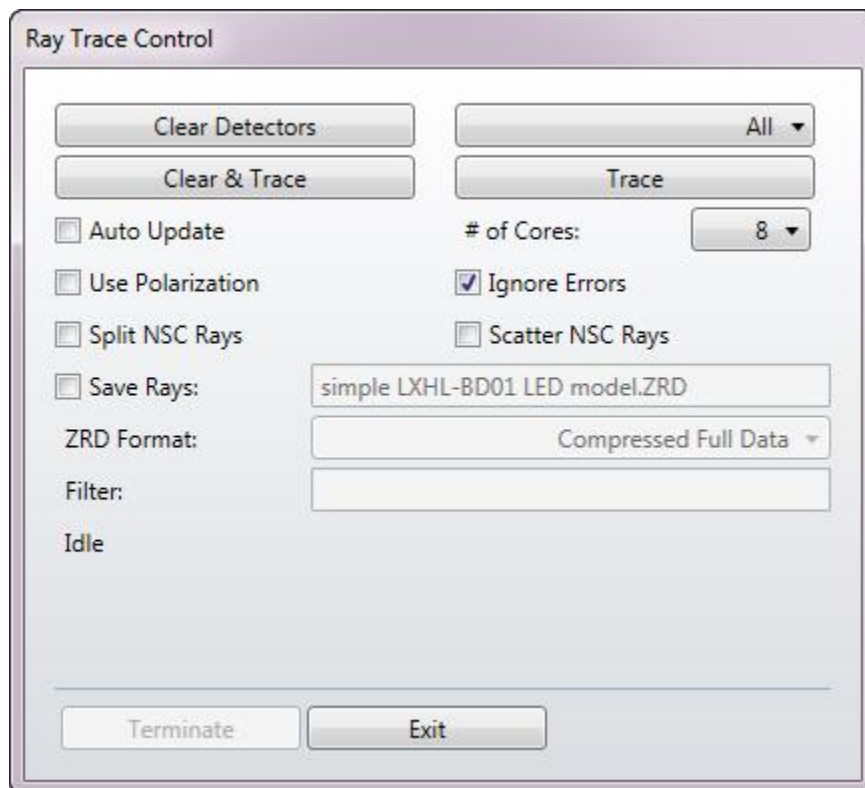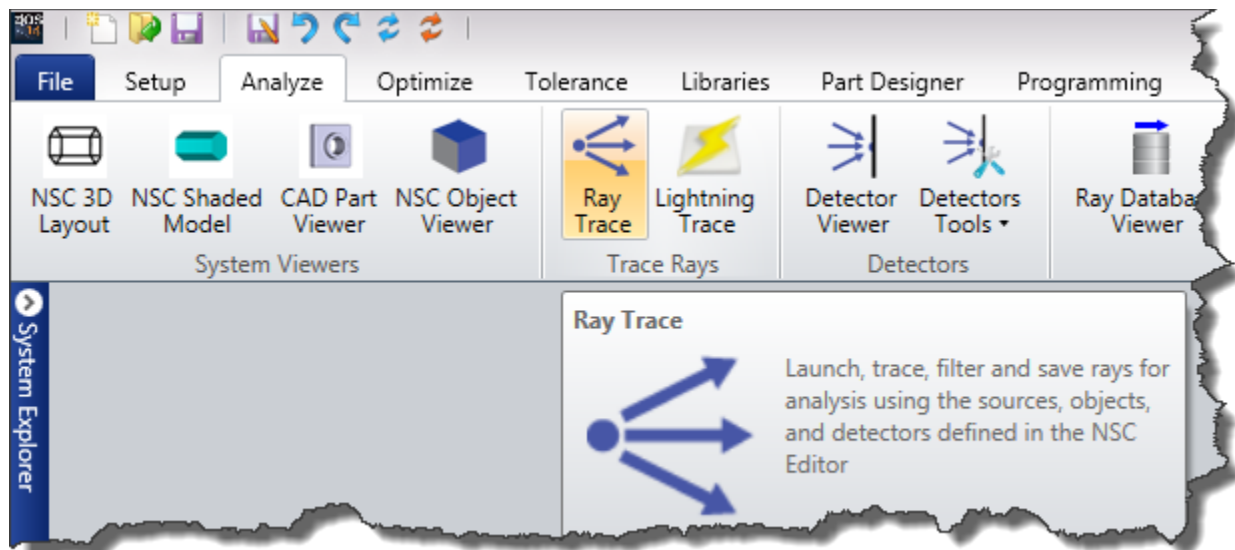
Click Clear and Trace to trace the million Analysis rays, then click Exit. If you have more than one CPU in your computer, the calculation will be automatically split up over all available CPUs.

Click on Analyze…Detector Viewer to see the data inside the Detector.
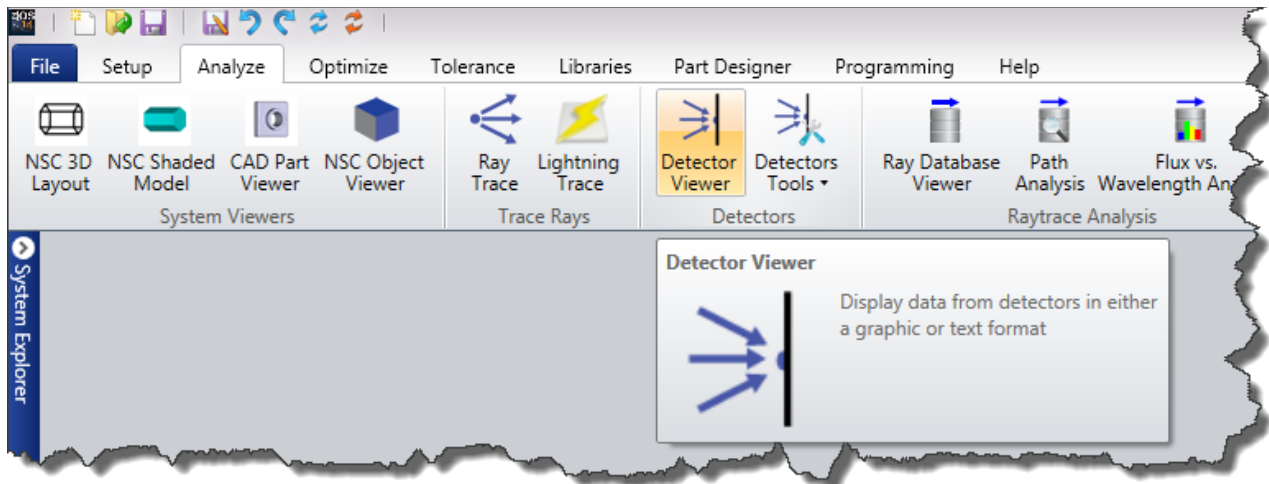
The Settings dialog for the Detector Viewer is very powerful, and allows you to select incoherent illuminance, Luminous intensity, coherent illuminance and phase (not meaningful in this case) and luminance. Multiple Detector Viewer windows can be open simultaneously to display multiple views of the same data.
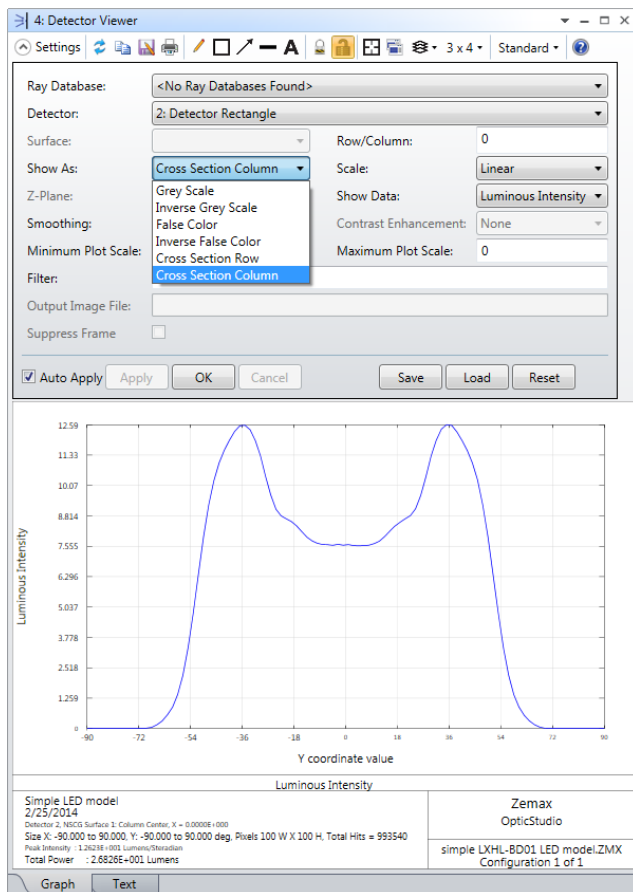
You can also show the data in false color, grey scale or look at cross-sections through the data using the 'Show As' control. When using cross-section views, row or column 0 always means the central row/column. Data can be scaled linearly or logarithmically.

The data can also be smoothed by averaging the data in each pixel and its neighboring pixels. The operation can be repeated the number of times specified by the smoothing parameter. This improves signal/noise at the expense of spatial or angular resolution.

Ray Databases

The Detector Viewers are very useful, but sometimes you will want access to the ray data directly. Repeat the ray trace (press the Rtc button) and select 'Save Rays'
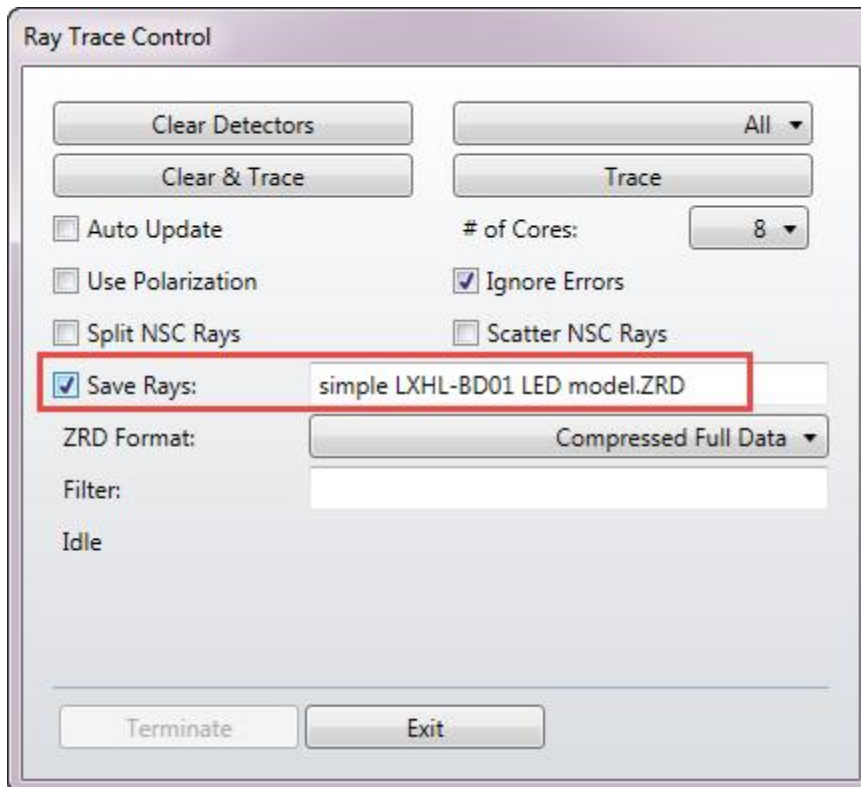
Note that the ray-trace takes a little longer now because of the time taken to write one million ray-histories out to disk. Then press Analyze...Ray Database Viewer:
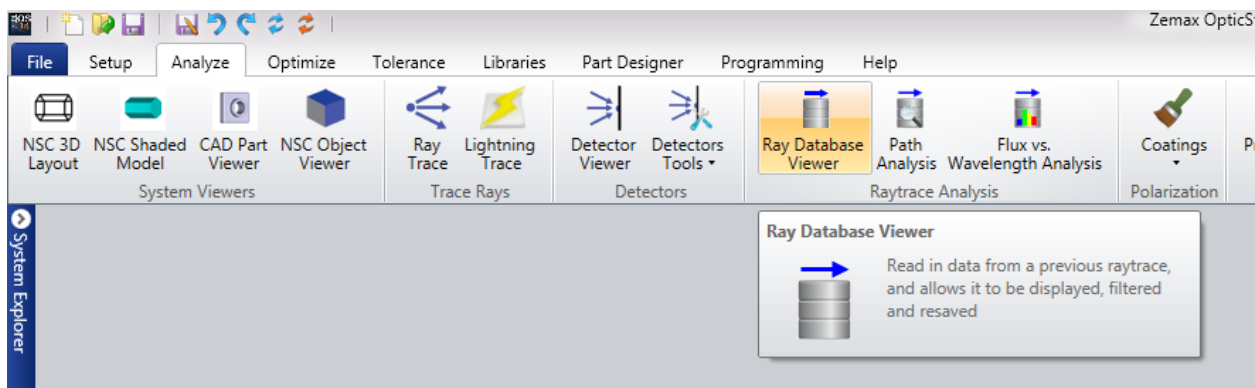
The Ray Database Viewer shows the history of every ray traced. The intensity position, direction cosines, normals, path length and polarization data of every ray can be shown, although only intensity is shown here. The ray is broken down

into segments, where each segment is a single ray-object intersection. Segment 0 is the ray data at the source. Various parameters XRTS etc. show what happened at the end of the segment (X= terminated, R= reflected, etc.). This example is very simple in that rays are launched, traced once and terminated:
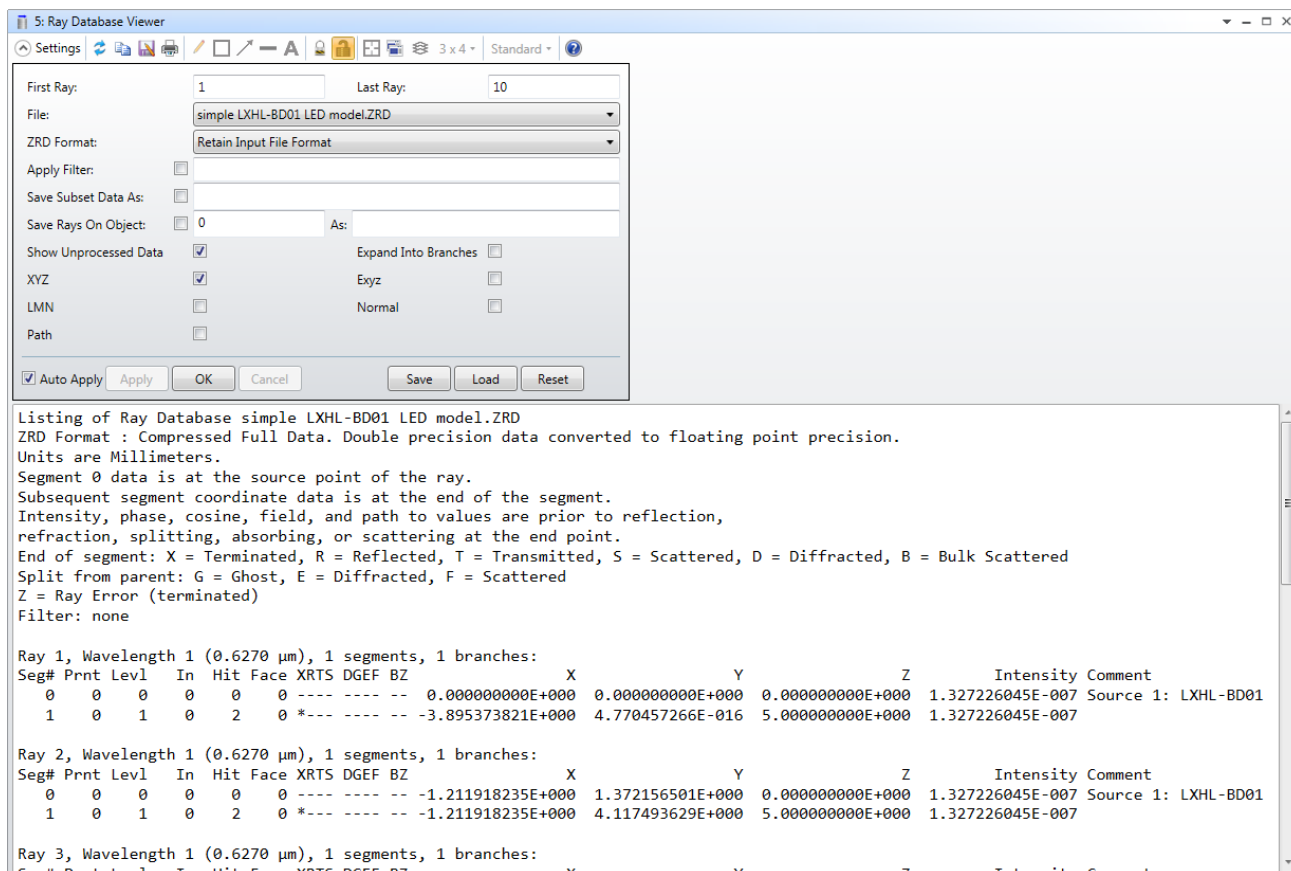
In a more realistic system, there are many more segments, of course. Load the sample file led_model.zmx (in the same folder as the file we are currently using), trace Analysis rays using these settings:
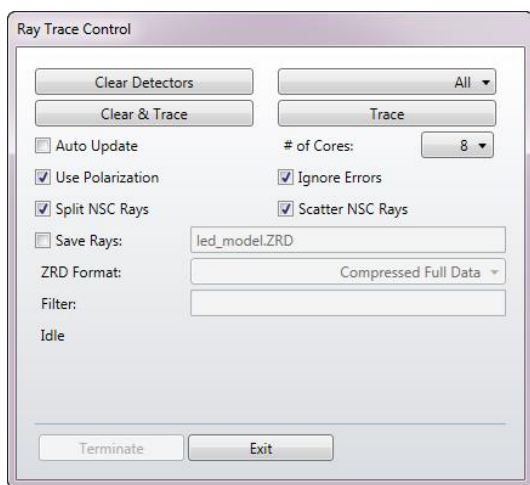
Use the Ray Database Viewer to view the ray histories. Note that as ray splitting is on, you can use the option 'expand into branches' to identify each child ray separately.

Filter Strings

Because OpticStudio knows the history of every ray it has traced, we can use filter strings to identify rays that meet specific conditions easily. For example, in the led model.zmx file, object 2 is a reflector behind the source. Some rays are fired forwards, and never see this mirror, while others move in the opposite direction, hit the reflector, and then travel forwards:
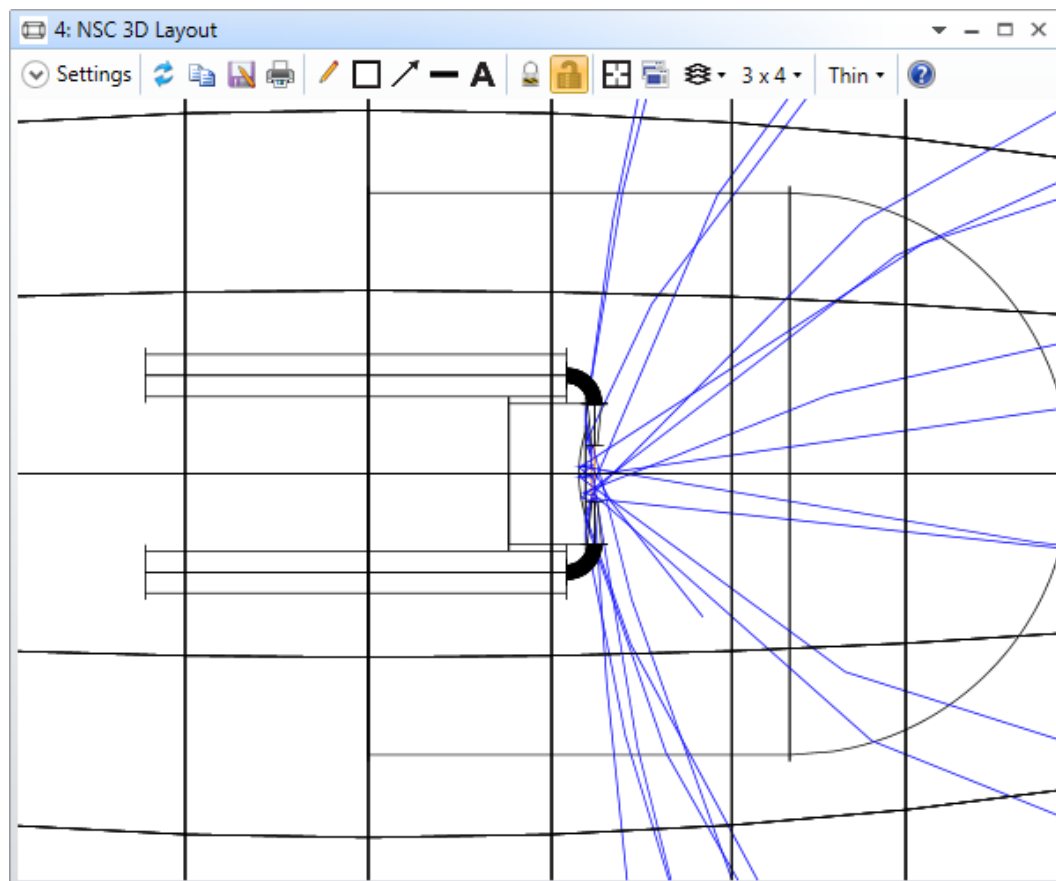
Using the filter string R2 means that only rays that reflect from object 2 will be shown:
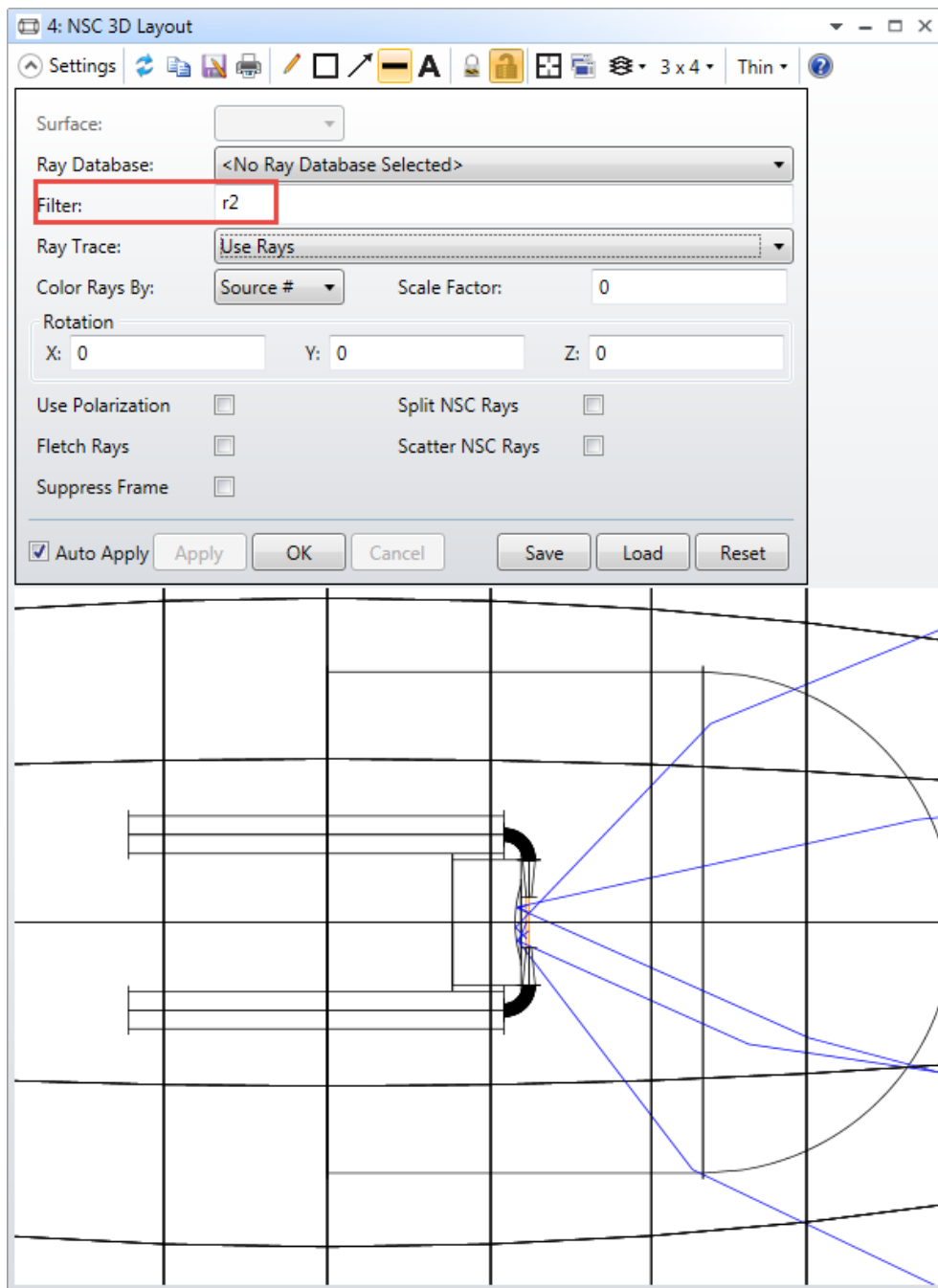
!R2 will show those rays that do NOT reflect from object 2, i.e. rays that propagate forwards initially. You can AND, OR, NOT, XOR, etc., multiple filters to produce a filter string that identifies exactly the conditions you want to investigate. For example, to select rays that must have either a) hit object 7 and object 9, but did not reflect off object 6, OR b) missed object 2, the filter string would be (H7 & H9 & !R6) | M2.

Filter strings are your most important tool for detailed system analysis. They can also be used with ray databases, both prior to saving the ray-database and with the saved data. For example, in stray-light simulations you may have to trace millions of rays to get one that finds its way to the detector. By saving to disk only those rays that hit the detector, you can produce a manageable data set for further study.

You can replay ray databases through the ray-database viewer, the layout plots, and the detector viewers, and add further filters to the filtered data. The ray database viewer will also let you filter a ray database and save the sub-set data into a new file.

Tutorial 6.5: Complex Object Creation

No matter how many objects we add to OpticStudio, you may still sometimes need an object that is not directly available. There are, however, ways to manipulate existing objects so as to create precisely what you need.

The Overlapping Objects Rule

When two or more objects occupy the same region of space, a simple rule applies. The properties of the common region are defined by whichever object is listed last in the Non-Sequential Component Editor.
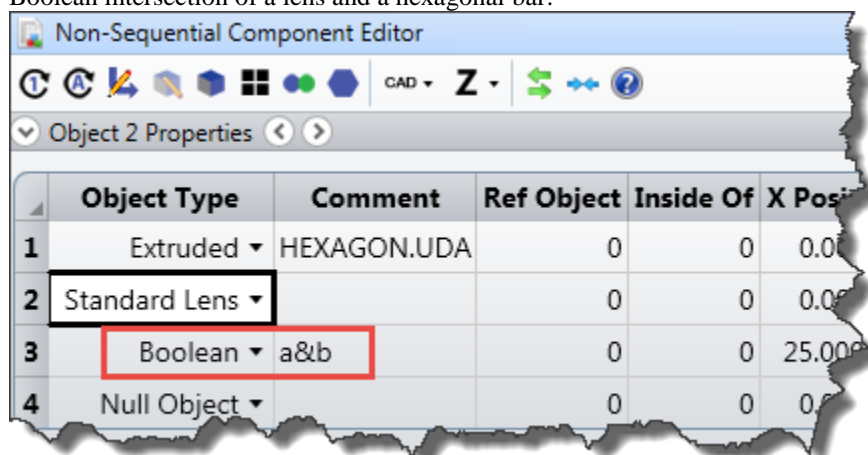
Open the sample file Non-sequential\Diffractives\Diffraction grating lens with hole.zmx. This mixed sequential/non-sequential design shows a lens with diffractive power, and a central region with no diffractive power:
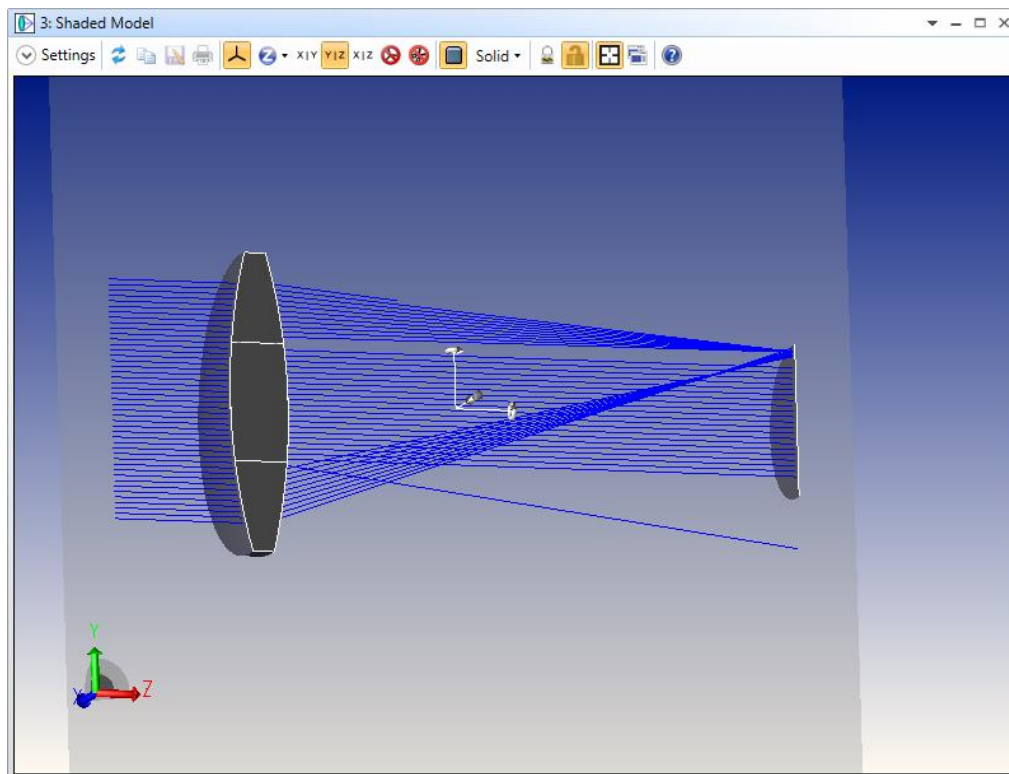
This is easily achieved by placing the non-diffractive element after the diffractive element in the editor, and co-locating them. There is no need to use the 'Inside Of' flag when geometry objects are nested inside each other, unless a source object is inside one of the nested objects. Geometry objects may fully or partially overlap, but source objects must always be entirely inside of any object they are co-located with. The 'Inside of' flag must be used for all nested geometry objects as well as sources in this case.

The Boolean Object

Up to eight objects may be combined in any order by the Boolean object, and Boolean objects may be combined with other objects, including other Boolean objects. For example, the sample file Samples\Non-sequential\Geometry Creation\Boolean Example 2- a lens with a hexagonal edge.ZMX shows how a hexagonal lens can be formed by the Boolean intersection of a lens and a hexagonal bar:

See any of the sample files in Non-sequential\Geometry Creation for further examples.

The Array Object

The Array object allows you to make one, two or three-dimensional arrays of any object, for example an array of Boolean objects (Samples\Non-sequential\Geometry Creation\Array Example 3- an array of Boolean objects.ZMX)
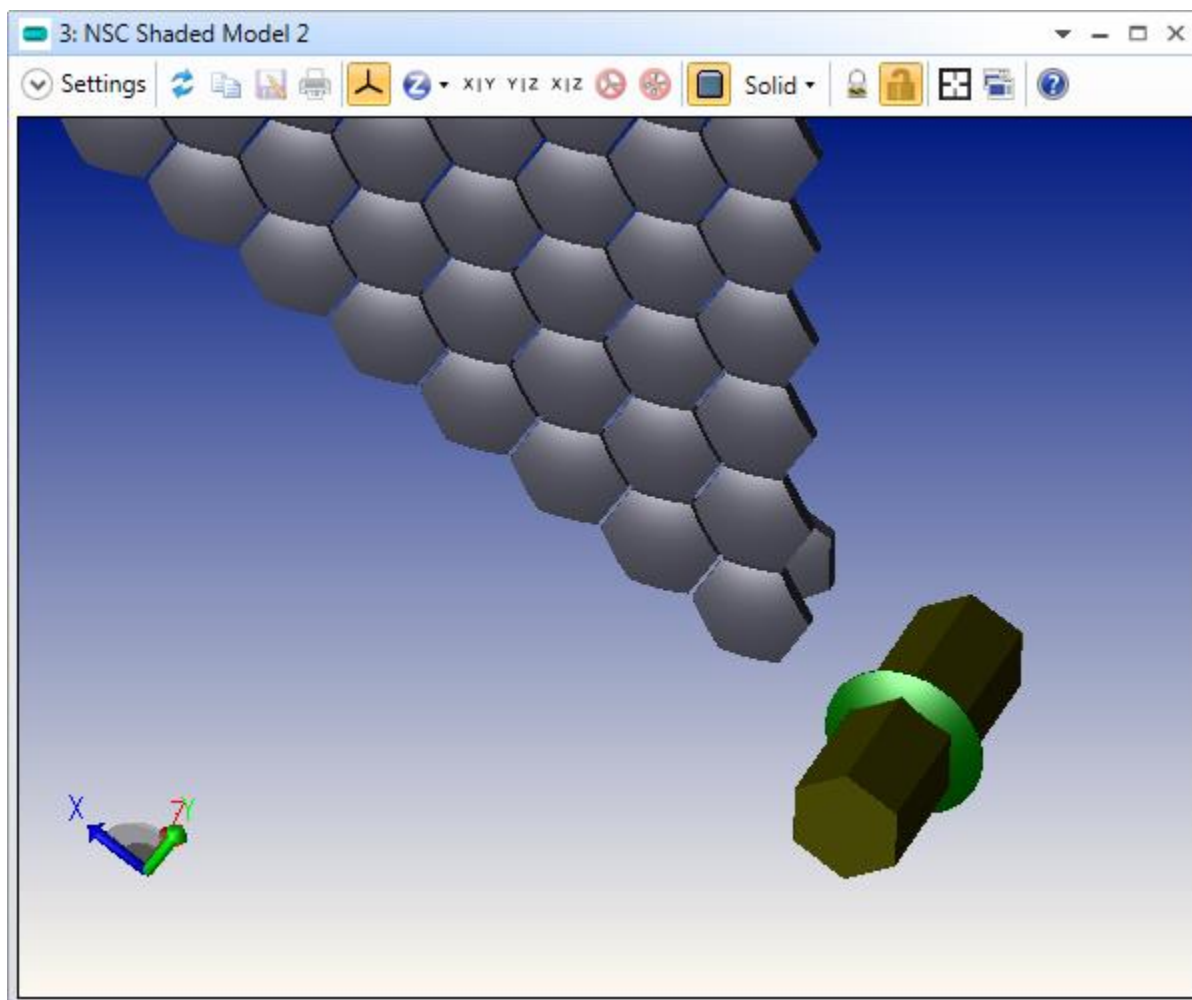
Using the Array object is highly recommended over the alternative method of multiple object definitions for several reasons:

- It uses much less memory than the equivalent number of individual objects: typically only slightly more than one instance of the parent object
- It employs sophisticated ray-trace acceleration techniques to trace orders of magnitude faster than the equivalent number of individual objects
- It is less error-prone than entering multiple objects, and only one object needs to be updated or optimized to update or optimize the whole array

The Array Ring can also be used to create circular, hexapolar, and spiral arrays. It has the same advantages as the Array object, outlined above. The sample below is Samples\Non-sequential\Geometry Creation\Ring Array Example 3- Hexapolar Array.ZMX
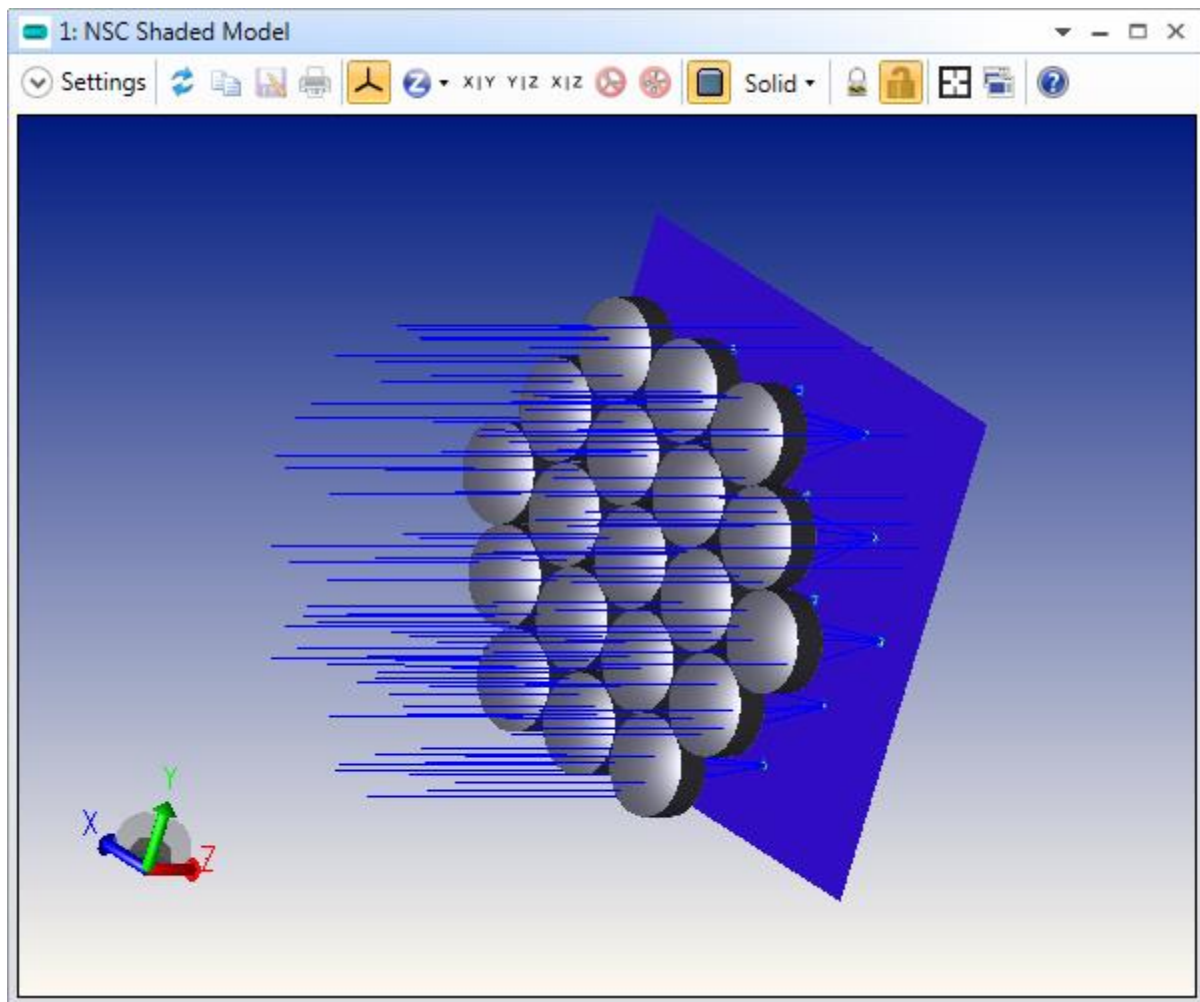
The Source Object

Any geometry object may be used as a source by using the Source object. This is ideal for infra-red and Narcissus analysis, where the emissivity of opto-mechanical components must be accounted for.

Tutorial 7: Optimizing Non-Sequential Systems

Optimization is fully supported in both pure non-sequential and hybrid non-sequential/sequential optical systems. The most common way to optimize a pure non-sequential system is via the Optimization Wizards, which can then be modified with specific requirements. OpticStudio supports three Wizards for non-sequential designs:
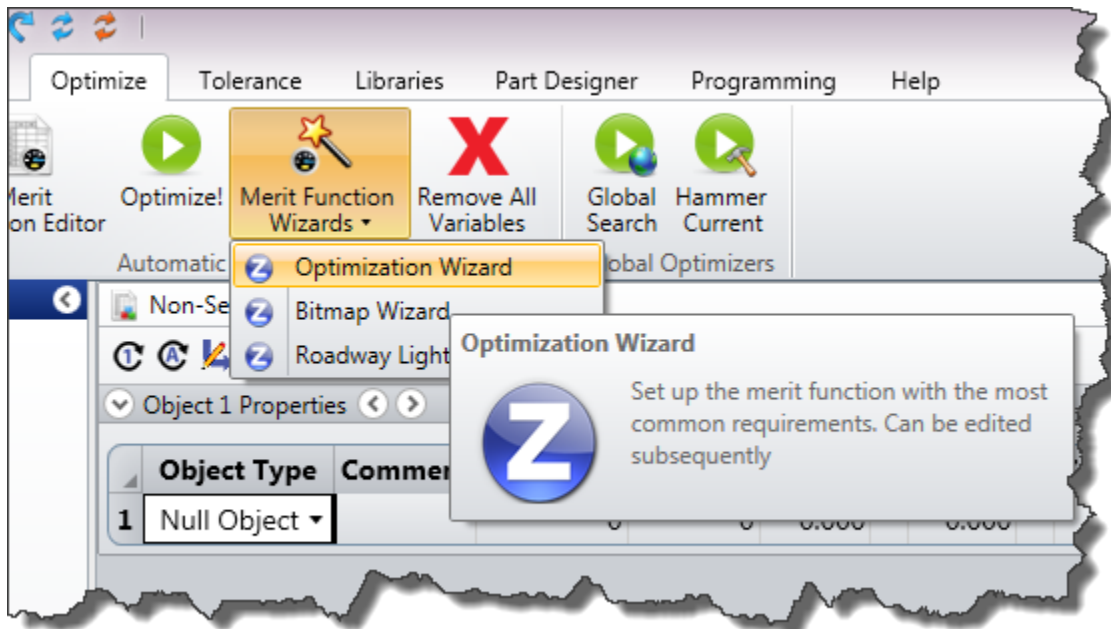
The most common operands used in non-sequential ray tracing are the NSTR and NSDD operands, although other operands are also available. NSTR is used to trace rays, and it acts exactly like the Ray Trace Control dialog. NSDD is used to clear detectors and to read out detector data. Any pixel can be read out directly, but for optimization it is usually more useful to optimize on aggregate ray data, like centroid location or effective widths in the spatial or angular domains. By using negative pixel numbers in the NSDD operand, OpticStudio will compute data like the average and standard deviation of all pixel data, and spatial or angular centroids and RMS widths.

Open the sample file in the Samples\Non-sequential\Miscellaneous folder called Freeform Optimization.zmx. This file contains a CAD part and source ray file supplied by Osram for their LB_T67c LED. It also contains a lightpipe, the shape of which we wish to optimize.

Note that although the data file for the LED contains 50.7 lm (lumen) of power, we have set the Source power to be 1 lm. This has no effect on the rays traced other than a scaling of intensity, but it makes it easier to discuss efficiency as the total power is now 1 lm.
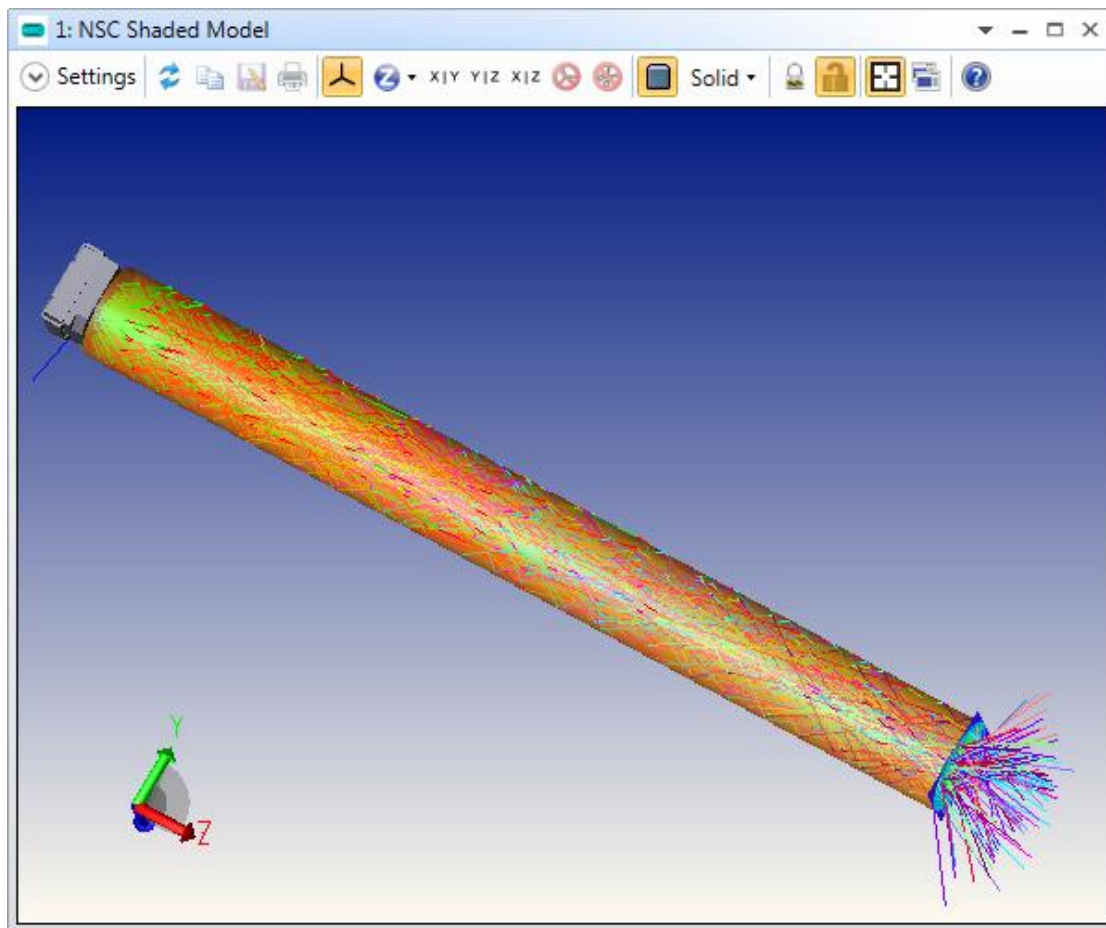
The lightpipe is a Freeform-Z object, which is defined by a set of (y, z) data points. OpticStudio fits a smooth curve through these data points, and then rotates the curve around the z-axis to form a rotationally symmetric pipe. The pipe is currently just a cylinder, but note that the (y,z) data is set to be variable.

Note also that the z-position, and x, y halfwidths of the detector object are locked to the Freeform-z object by pickup solves. These ensure that the detector will always be just in front of the output face of the lightpipe, as the pipe's length is varied during optimization, and adjust the width of the detector so it always captures all the light from the output face as the width of the output face changes.

The primary goal for this light pipe is that it should give the highest brightness in the forward direction. Therefore its luminous intensity should be as high as possible and the width (in angle space) of the luminous intensity plot should be as small as possible. In addition, there are some mechanical constraints on the maximum and minimum widths and length of the lightpipe that must be met.

Open the Merit Function Editor and examine the merit function. First the detectors are cleared, and then rays are traced. Then we compute the RMS angular width of the detector data by using pixel -9 (which is RMS width) and Data = 2 (power/unit solid angle). The starting beam has an RMS angular width of 48.5°:

and a peak luminous intensity of 0.38 lm/sr, although this is clearly a noisy number:
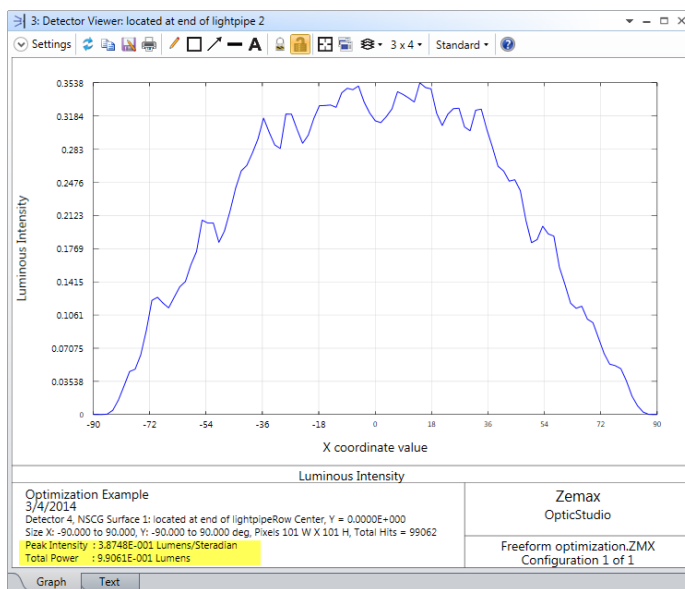
In addition, we also target the total power detected to be as large as possible. This is an important constraint, because if no rays land on the detector, the RMS angular width is identically zero! This is not a solution we want, so we are optimizing for maximum received power and minimum angular width.

There are also some constraints on the shape of the pipe, and you should consult the full description of the FREZ operand for full details. These constraints prevent the lightpipe from becoming unfeasibly thick or thin.

Run the optimizer, and select the 'Orthogonal Descent' optimization operand. This alternative local optimizer is very good at making big improvements quickly, especially in non-sequential systems, although the Damped Least Squares optimizer can usually make further improvements on it.

After five cycles of optimization, the lightpipe's shape has evolved to produce a peak luminous intensity of 102 lm/sr, which is over 200 times brighter than the starting design, and an RMS angular radius of less than 9°:
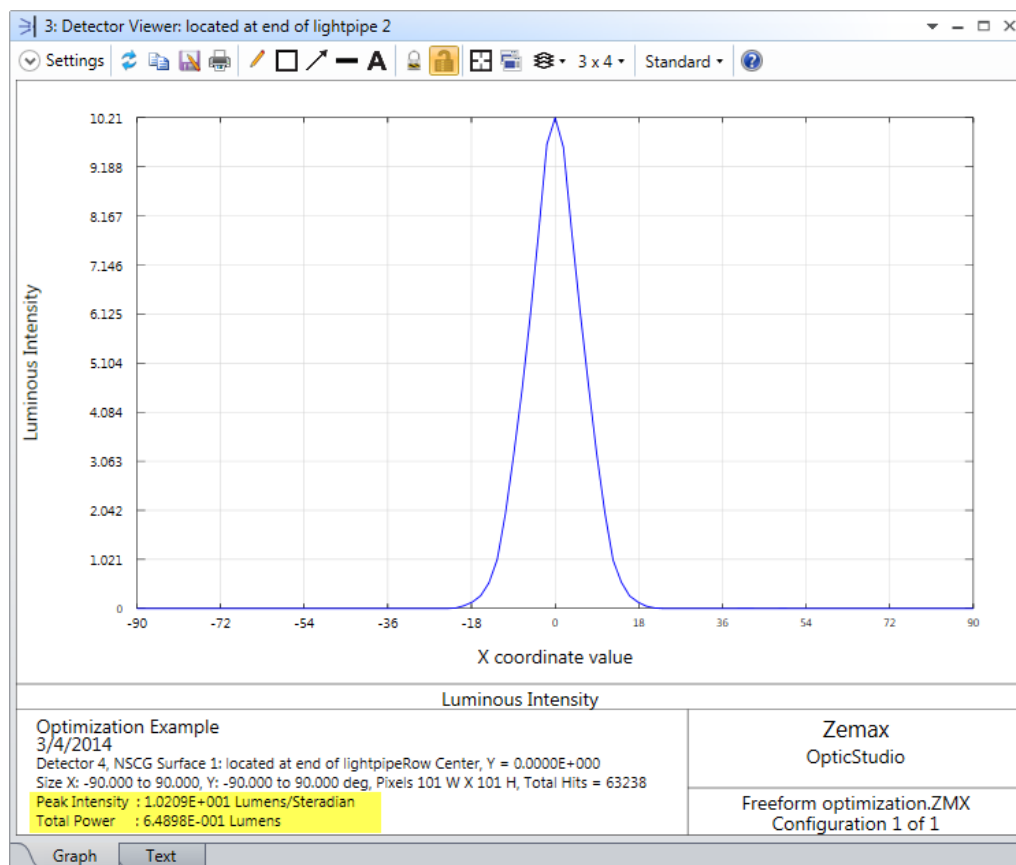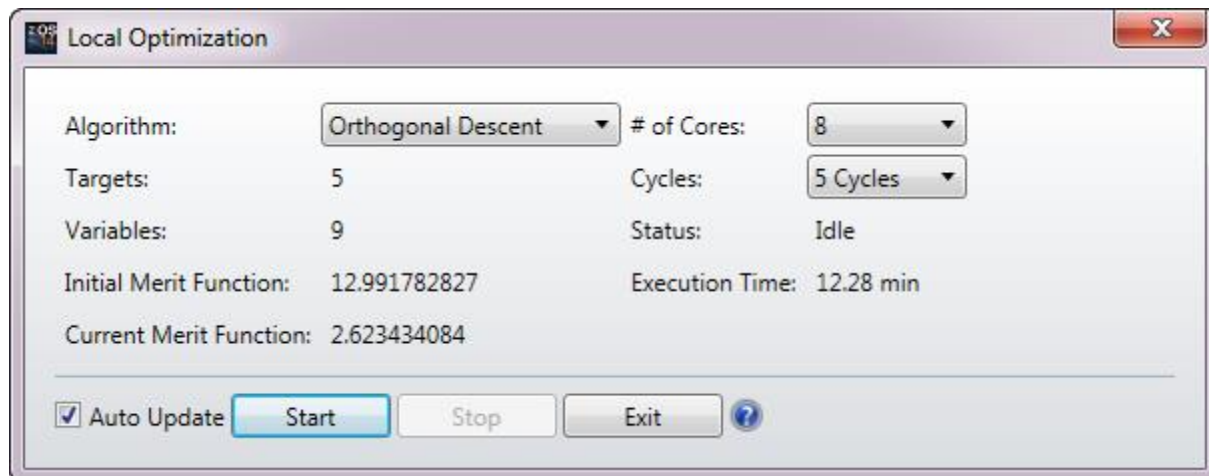
Note also that the detector object has shifted position and increased in size because of the pick-up solves used, and that the pipe is ~65% efficient (check the total power above and compare to our 1 lm launch power)

Tutorial 8: Colorimetry

Colorimetry is the study of color, which is the response of the human eye to optical radiation in the wavelength range 0.38 to 0.83 microns. The color of any non-sequential source object can be defined in many ways.

If the wavelength spectrum of a source is known, and there is only one such spectrum it can be entered directly in the System Explorer's Wavelength section (up to 24 data points). However, this is uncommon in colorimetry are other methods are more commonly used. In the Libraries tab, Source Viewers group, select the Source Spectrum Plot and look at its Settings:
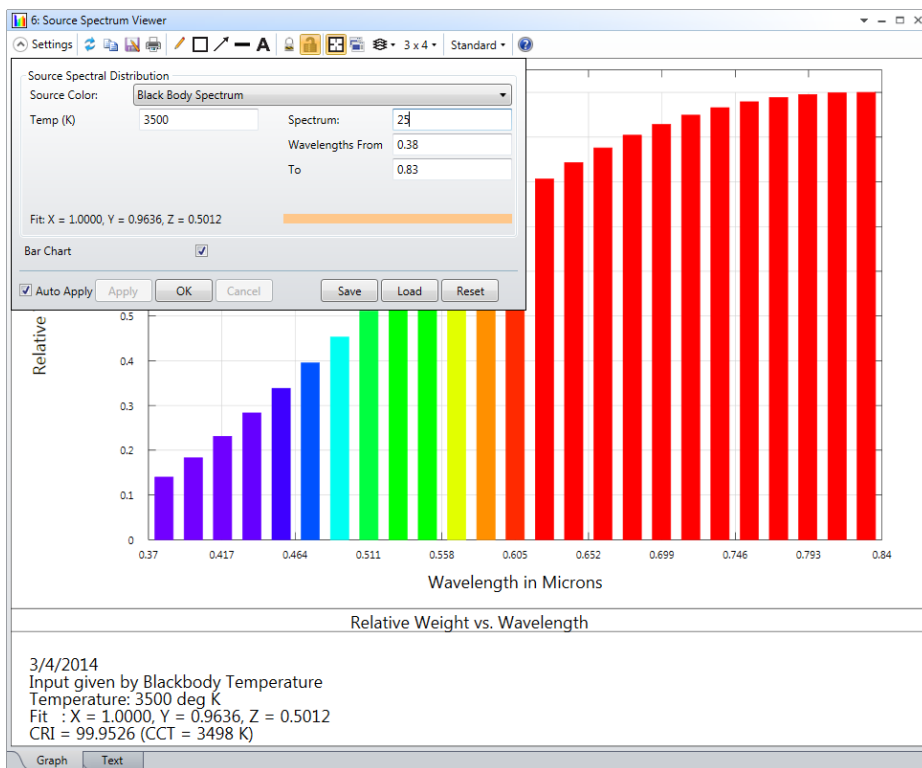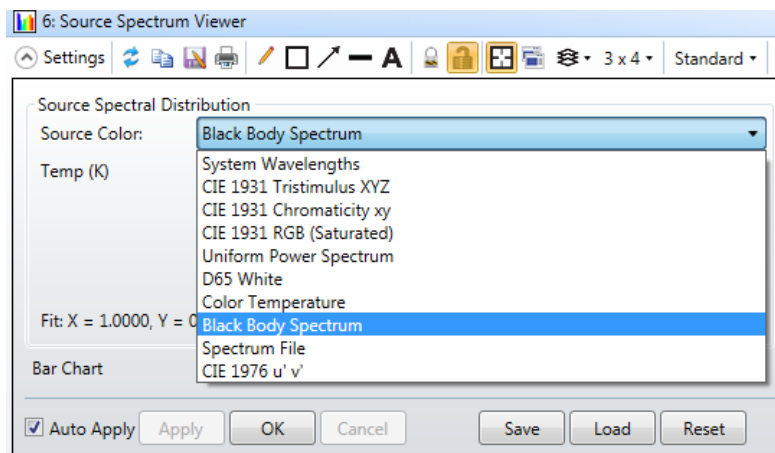
The viewer is useful for investigating all the color models that OpticStudio supports, including those fitted from CIE color coordinates. Once you know how to define the colors you want, you enter this data within the Sources section of the Object Property Inspector:
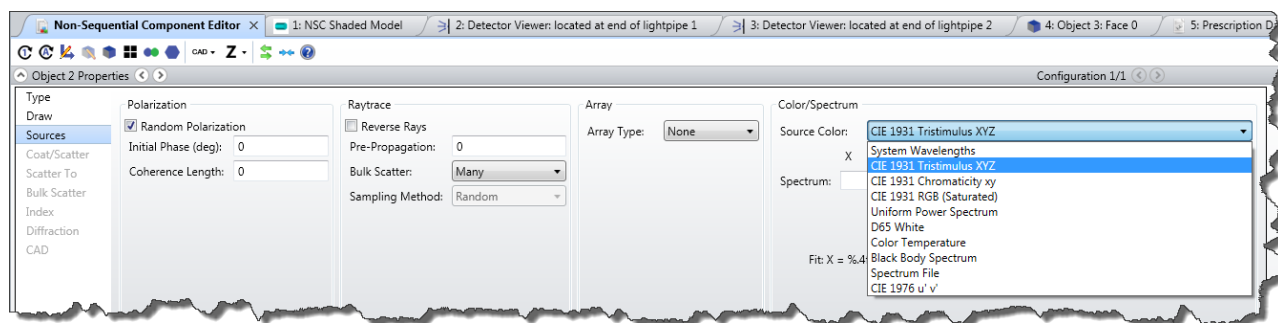
Any number of sources, each with its own unique color, can be defined. For example, if you have CIE 1976 u'v' data for a source where u' = .31 and v' = 0.5, OpticStudio can fit this color exactly using just four wavelengths:
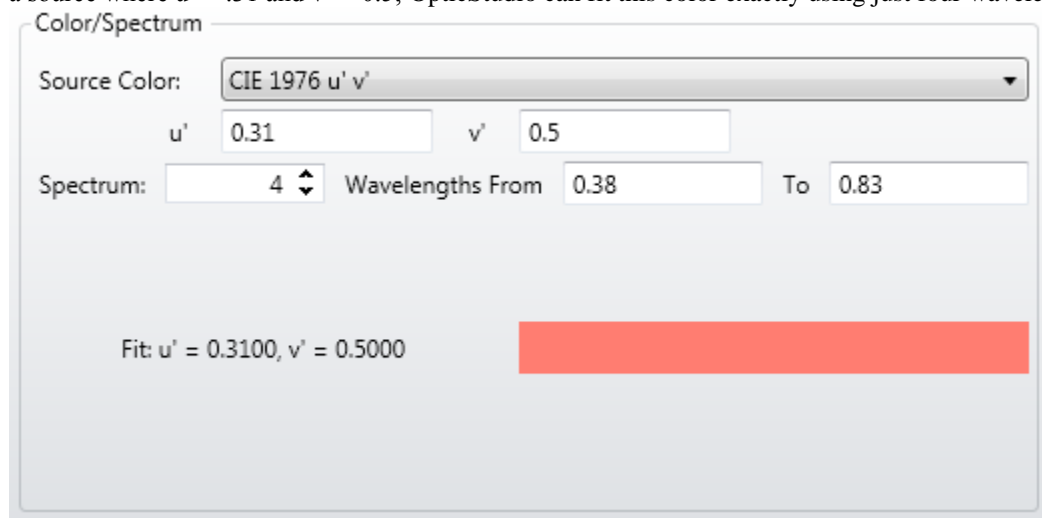
In general, you should use the minimum number of wavelengths that gives adequate color rendering. Rays are then traced using either the specified or synthesized spectra of each source, until they are detected by a Detector Color or Detector Polar object which can provide either True Color (photometric) or False Color (radiometric) data as required by the user.

For example, open the sample file sub-folder \Colorimetry in the non-sequential samples folder, and open Example 1, two color mixing gives white. White LEDs can be produced by using two phosphors in the LED die, giving spectra in the blue and yellow. For simplicity, in this file we show the blue and yellow sources as being separate, and also as overlapping:
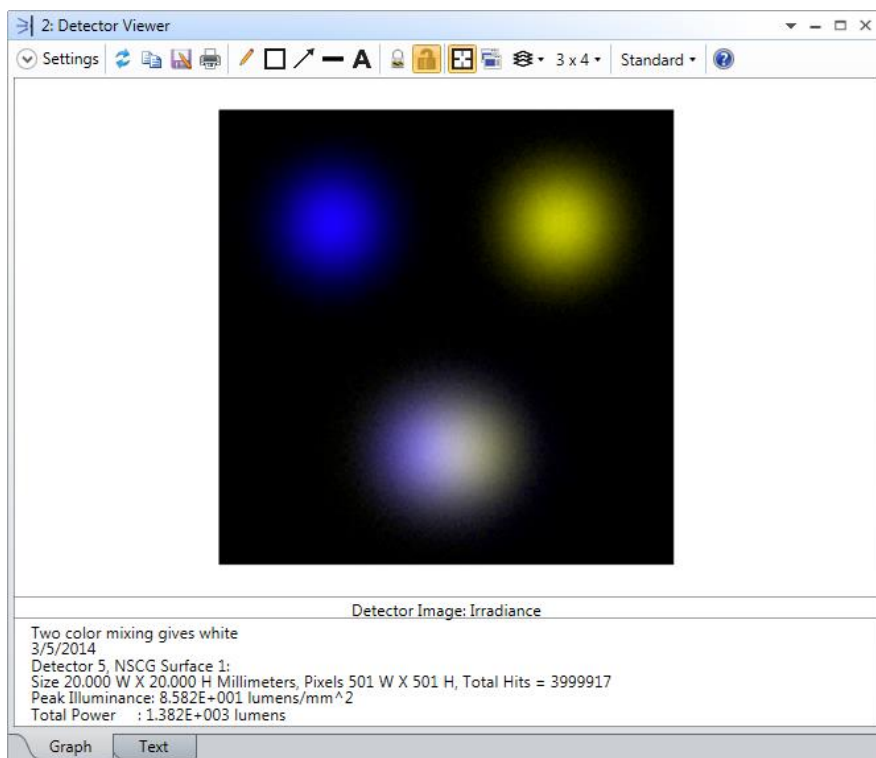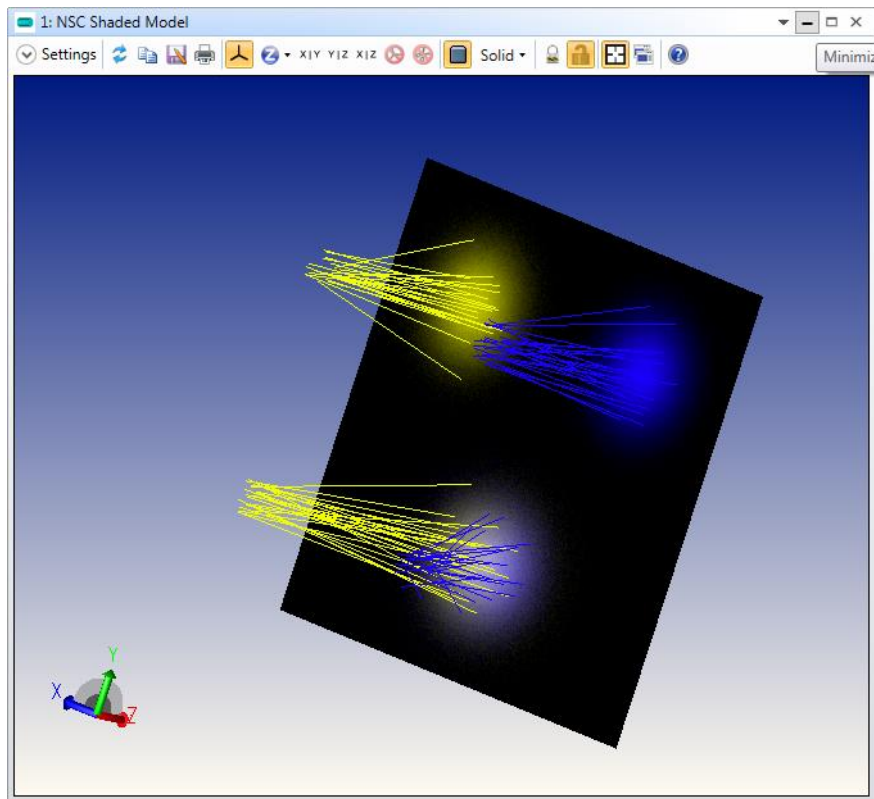
Note that because the two beams do not overlap perfectly, you can see a blue tinge on one side of the white spot, and a yellow tinge on the other. Optimization operands are available to allow each pixel's color to be analyzed and targeted in the merit function, so that you can optimize for a desired color or CRI.

Tutorial 9: Polarization, Coatings & Scattering

Ray tracing programs generally treat rays as purely geometric entities, which have only a position, orientation, and phase. For example, a ray is completely described at a surface by the ray intercept coordinates, the direction cosines which define the angles the ray makes with respect to the local coordinate axes, and the phase, which determines the optical path length or difference along the ray.

At the boundary between two media, such as glass and air, refraction occurs according to Snell's law. Usually, the effects at the interface which do not affect beam direction are ignored. These effects include amplitude and phase variations of the electric field which depend upon the angle of incidence, the incident polarization, and the properties of the two media and any optical coatings at the interface.

Polarization analysis is an extension to conventional ray tracing which considers the effects that optical coatings and reflection and absorption losses have on the propagation of light through an optical system. It is available for sequential and non-sequential systems.

Further, scattering at the interface can also be considered. Scattering is due to micro-structure of the surface texture: at a sufficiently fine resolution, the surface of a 'smooth', polished glass is really a rough surface, with the result that the departing direction cosines are perturbed, or scattered, about their specular values. Scattering can also occur during ray-tracing through an optical material, due to inclusions in the material. This is referred to as 'bulk scattering'.

Tutorial 9.1: Polarization

In addition to position and direction, the amplitude and polarization state of a ray can be described by a vector $\underline{\mathbf{E}}$ with complex valued components (E_x, E_y, E_z). Since the $\underline{\mathbf{E}}$ vector must be orthogonal to the ray direction vector $\underline{\mathbf{k}}$, (given by the (l, m, n) direction cosines of the ray) then $\underline{\mathbf{k}}.\underline{\mathbf{E}} = 0$ and

$$E_x.l + E_y.m + E_z.n = 0$$

Since we know the direction cosines, we only need to specify the complex values of E_x and E_y, as E_z is then defined. The polarization can then be defined using a 2D Jones vector $\underline{\mathbf{J}} = (J_x, J_y)$ where J_x and J_y are measured _along the direction of the ray_ and have both a magnitude and a phase. The 3D $\underline{\mathbf{E}}$ vector is then constructed from the 2D $\underline{\mathbf{J}}$ vector and the direction cosines of the ray.

The method used to define the initial polarization of a ray then depends on whether we are working with a sequential or non-sequential system.

Defining Polarization in a Sequential System

The default polarization state of rays is defined in the General dialog box's Polarization section:
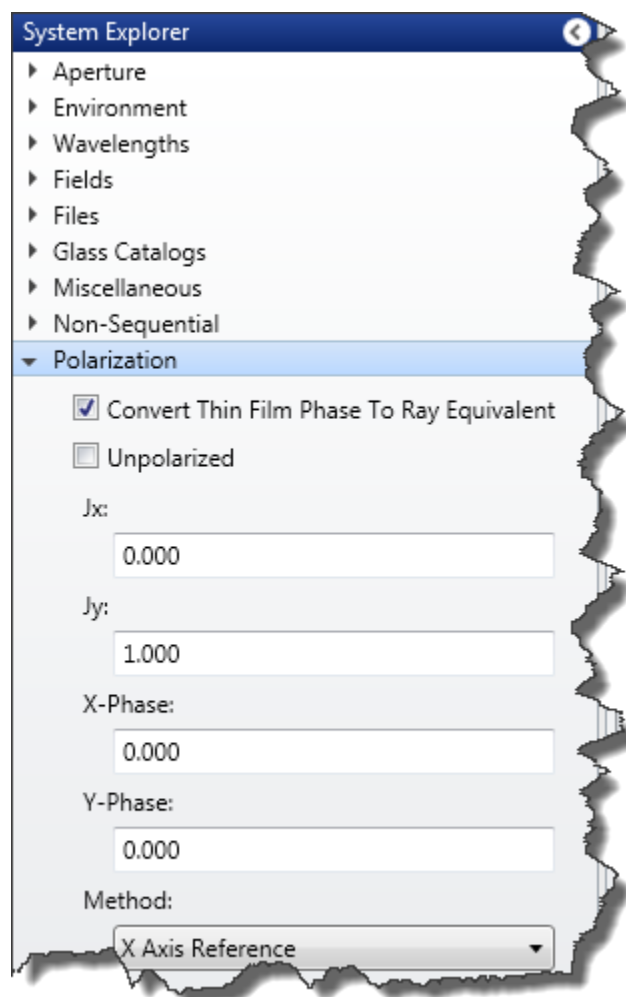
If an Analysis feature uses polarization, but its Settings do not allow for the definition of the polarization, then that calculation will use the settings here. However, many Analysis windows do allow the direct definition of the polarization state. These are the ones in the special 'Polarization and Surface Physics' group:
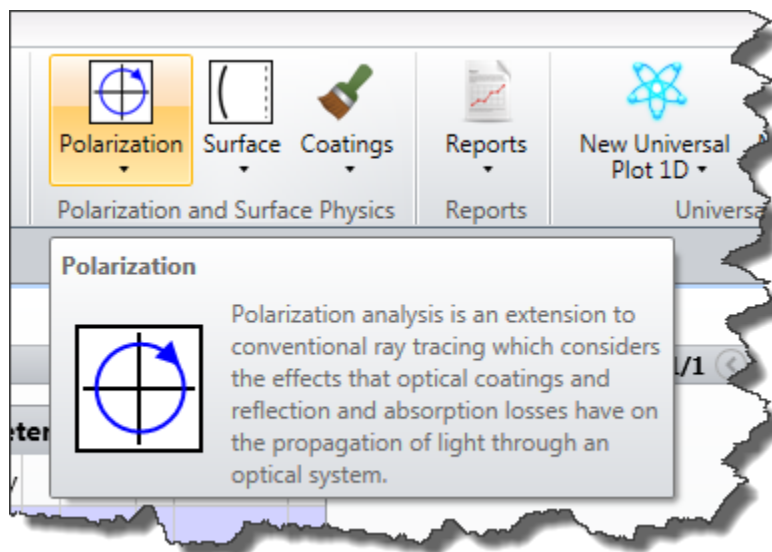
These windows *default* to the setting in the System Explorer, but allow direct modification. For example, the Polarization Ray Trace lets you can define the ray coordinates and polarization state directly:

4: Polarization Ray Trace ✕

Settings

Jx:	0	Hx:	0
Jy:	1	Hy:	0
X-Phase:	0	Px:	0
Y-Phase:	0	Py:	1
Global Coordinates: ☐		Wavelength:	2 ▾

☑ Auto Apply Apply OK Cancel Save Load Reset

...equential\Objectives\Double Gauss 28 d...

```
Normalized Y Field Coord (Hy) :     0.000000000
Normalized X Pupil Coord (Px) :     0.000000000
Normalized Y Pupil Coord (Py) :     1.000000000

Input Polarization:
X-Field   :   0.000000
Y-Field   :   1.000000
X-Phase   :   0.000000
Y-Phase   :   0.000000

All coordinates and cosines are in surface local coordinates.



Tracing ray to surface 1:

Path length through air              (tau):  2.6279917E+000
Internal absorption per mm         (alpha):  0.0000000E+000
Internal Transmittance of ray         (IT):  1.000000000000
Propagation Phase Factors         (pc,ps): -0.864366762953 -0.502861908582
Coordinates on surface            (x,y,z):  0.0000000E+000   1.6665000E+001   2.6279917E+000
Direction cosines of incident ray (l1,m1,n1):  0.000000000000   0.000000000000   1.000000000000
Cosine of angle of incident ray        :  0.951471206067 (17.922952 deg)
```

Defining Polarization in a Non-Sequential System

Non-Sequential source objects can have their polarization state defined in the Property Inspector Sources section:
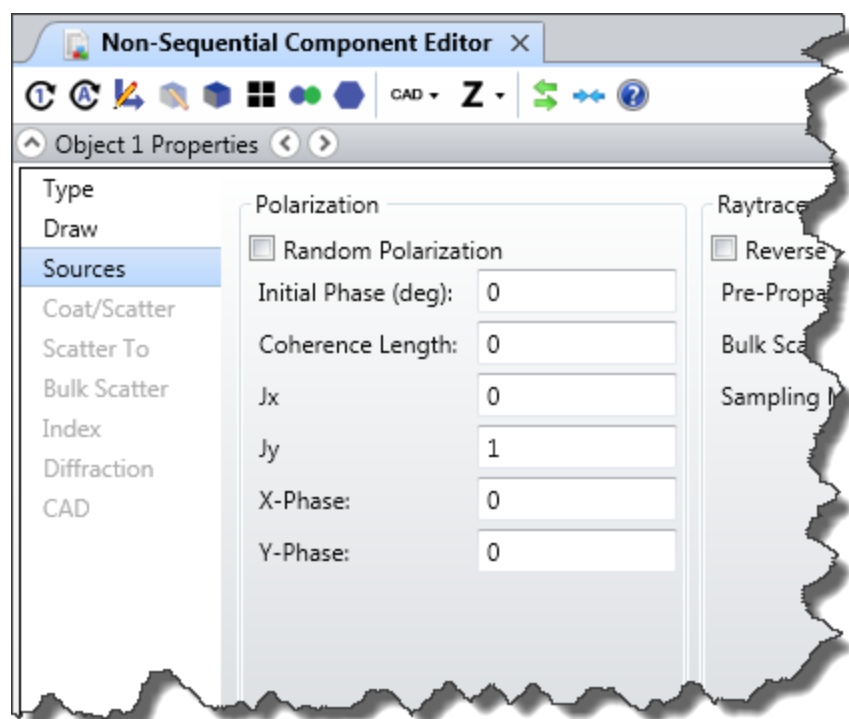
Every source can have separate polarization properties specified for it.

Tutorial 9.2: Thin-Film Coatings

OpticStudio has an extensive thin film modeling capability to support the polarization analysis. Multilayer film dielectric and metallic coatings may be defined, from either a predefined or user defined material database. Many thin-film codes, like The Essential Macleod, TFCalc and Film-Star, export coating designs directly in OpticStudio format.

Coatings may be applied to either dielectric or metallic substrates. Coatings may be composed of arbitrary layers of arbitrary material, each defined with a complex index of refraction, with full dispersion modeling in the coating materials. Substrates may be glass, metallic, or user defined. Coating layers may be of uniform or varying thickness, and loops of replicated coating stacks can be easily created.

OpticStudio automatically reverses the coating layer order if surfaces go from air to glass then glass to air, so the same coating may be applied on many surfaces without the need to define "mirror image" coatings.

Coatings are defined in a file with the .dat extension. This file is located in the coatings folder, which by default is My Documents\Zemax\Coatings. This folder can be modified by clicking on Project Preferences in the Setup tab. OpticStudio is shipped with a file called coating.dat, which contains sample data.

Note: You should not edit coatings.dat, as it is provided by the OpticStudio installer and will be overwritten when you next install an update.

Always place your own coating data in your own .dat file, and load it via the System Explorer Files section:
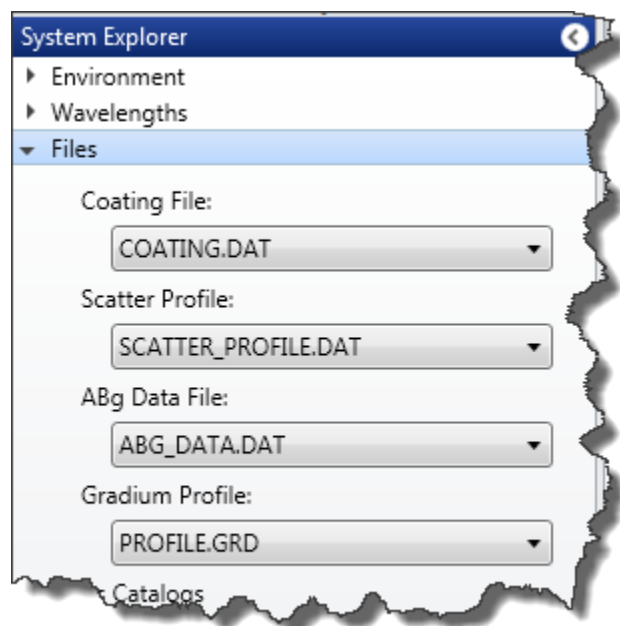
With the coating data in place, OpticStudio computes the diattenuation, phase, retardance, reflection, transmission, or absorption of any coating as a function of wavelength or angle.

Adding Coatings to Sequential Surfaces

Open the sample file Sequential\Objectives\Double Gauss 28 degree field.zmx again. This uses the supplied coatings.dat file. Note the Coatings column in the Lens Data Editor:

Now go to the Libraries tab, and select the Coating Catalog icon from
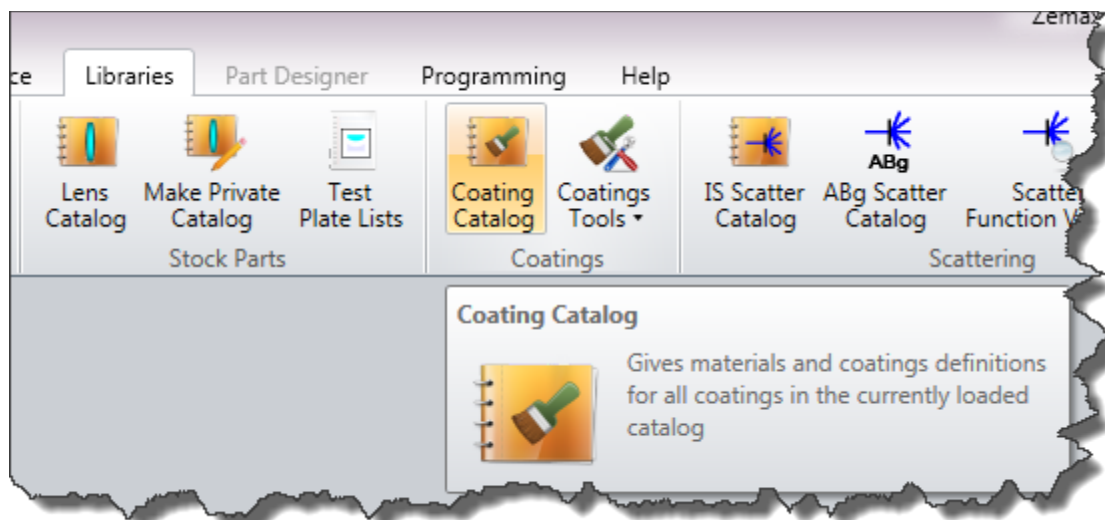
Scroll down to find the coating AR:
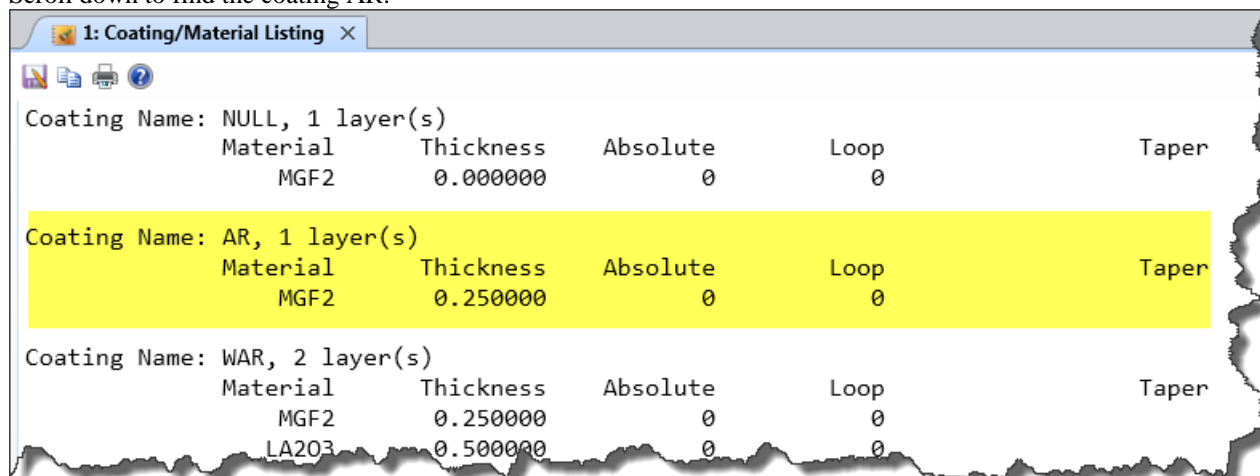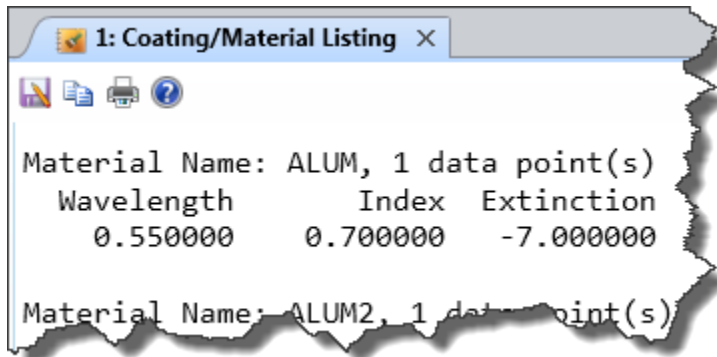
Coating thickness is given in units of waves at the primary wavelength unless the "Absolute" flag is non-zero; in which case the coating thickness is in µm independent of wavelength. The coating AR is, therefore, a λ/4 thick layer of the material MGF2, which is defined earlier in the coating.dat file:

Material Name: MGF2, 8 data point(s)
```
  Wavelength        Index   Extinction
    0.400000     1.383870     0.000000
    0.460000     1.381100     0.000000
    0.500000     1.379780     0.000000
    0.700000     1.376080     0.000000
    0.800000     1.375060     0.000000
    1.000000     1.373580     0.000000
    2.000000     1.367840     0.000000
    2.500000     1.364260     0.000000
```

This listing gives the complex refractive index, defined as $\eta = n + ik$, where n is the usual index of refraction, and k is the extinction coefficient. As the material MGF2 is defined as having positive n and zero k, it is a pure dielectric. The material ALUM however:

Material Name: ALUM, 1 data point(s)
```
  Wavelength        Index   Extinction
    0.550000     0.700000    -7.000000
```

Material Name: ALUM2, 1 data point(s)

has index <1 and negative extinction, and so is a metal (the OpticStudio convention is that extinction is negative for an absorbing medium). Coatings can be made up of any number of dielectric and metallic layers, the layer thickness can be constant or tapered, and repetitive loops of coatings can be easily defined.

If you do not have the coating prescription, OpticStudio supports several IDEAL coatings which allow you to just specify reflection and transmission, and also TABLE coatings that are similar to IDEAL coatings, except the transmission and reflection may be a function of incident angle and wavelength and may be specified separately for S and P polarizations.

Note: See the Polarization section of the Technical Reference for full details of the coating file syntax.

Now click on Coatings…Reflection vs Angle in the Analyze tab to see the performance of this coating on an SK2 substrate:
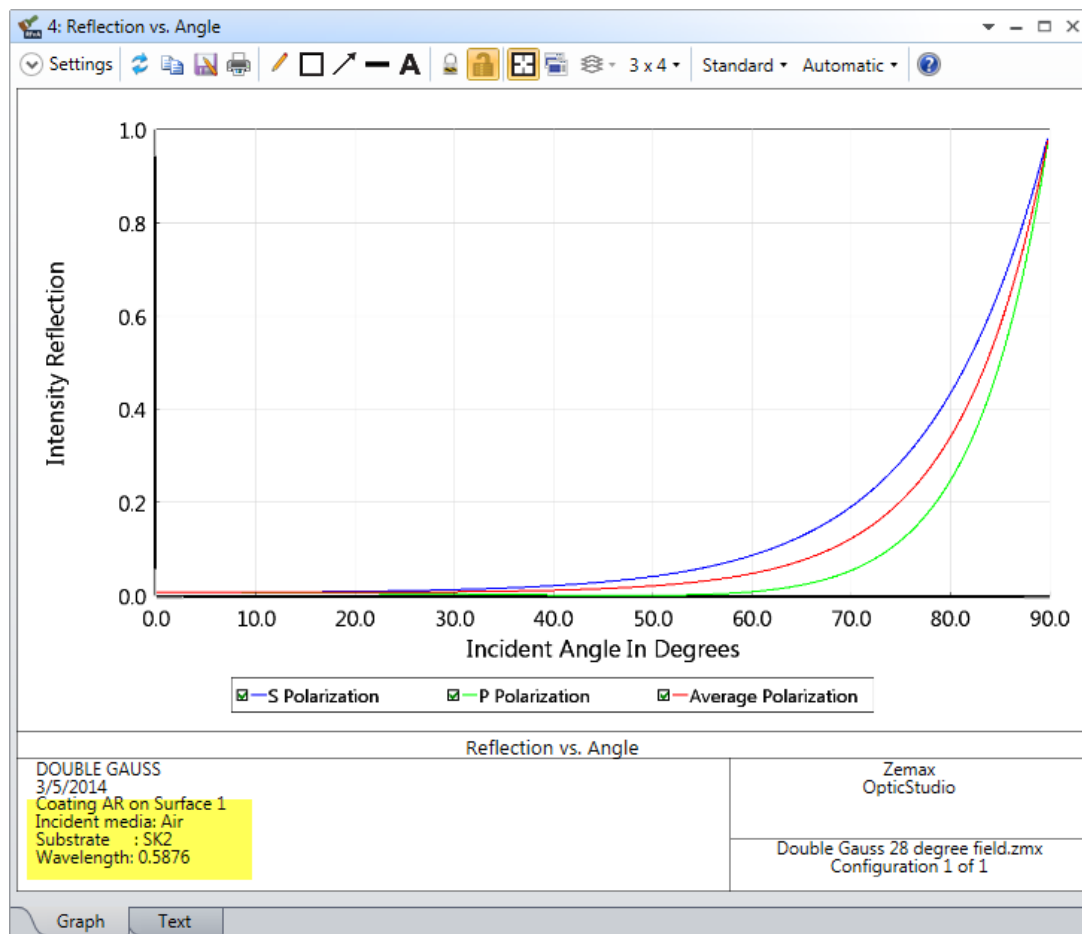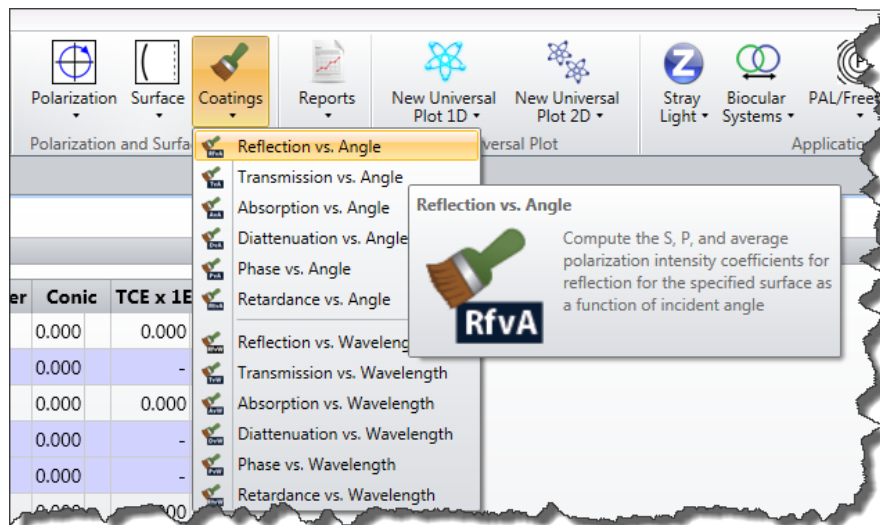
Other plots in the same menu let you see transmission, diattenuation, phase, retardance, etc. In the coating column of surface 1, press the spacebar to delete the coating, and note how the reflection vs. angle plot changes.

Coating names can be typed directly into the coating column to apply the coating to the surface, and the Apply Coating to All Surfaces tool in the Lens Data Editor toolbar lets you quickly add a coating to all surfaces of the system.
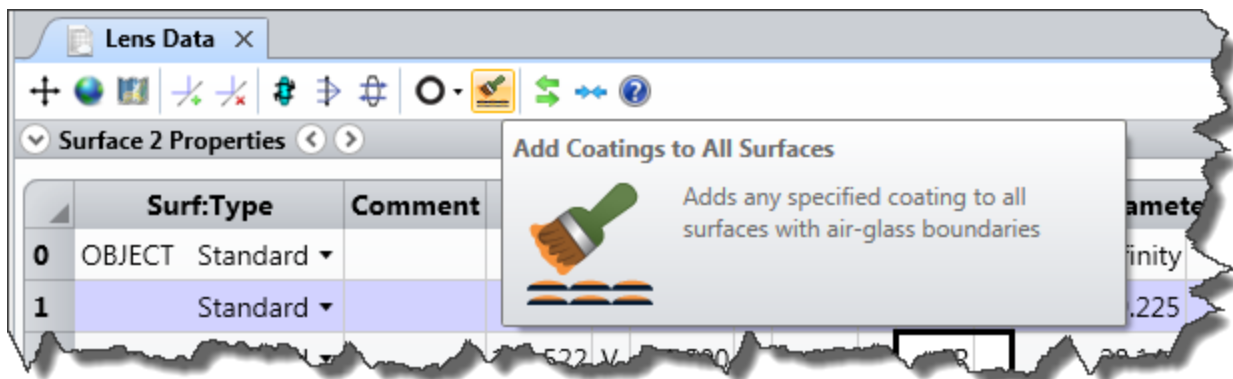
Adding Coatings to Non-Sequential Objects

It is only a little more complex to add coatings to non-sequential objects. Because objects are volumes and not surfaces, one object may have several faces that can take different coatings. Open the sample file Non-sequential\Ray splitting\Beam Splitter.zmx to demonstrate this:
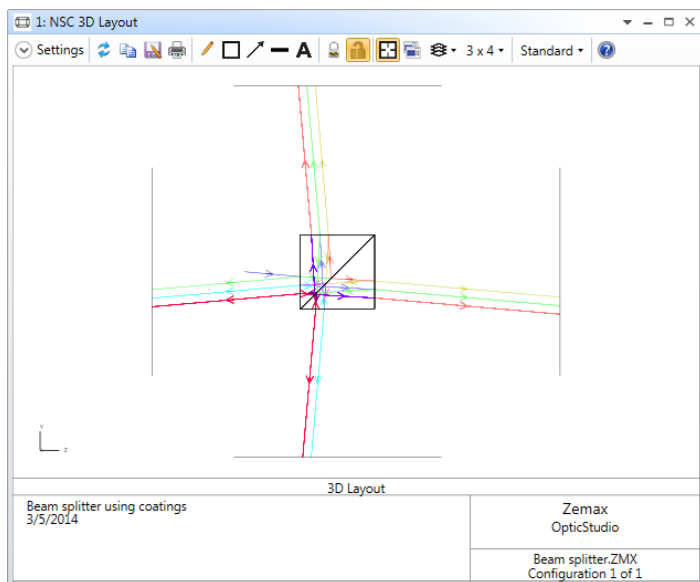
Double-click on the second prism object, object 3, to open the Property Inspector and go to the Coat/Scatter tab. This object has two faces: face 1 which is the splitter face (the hypotenuse of the prism) and face 0 is everything else.
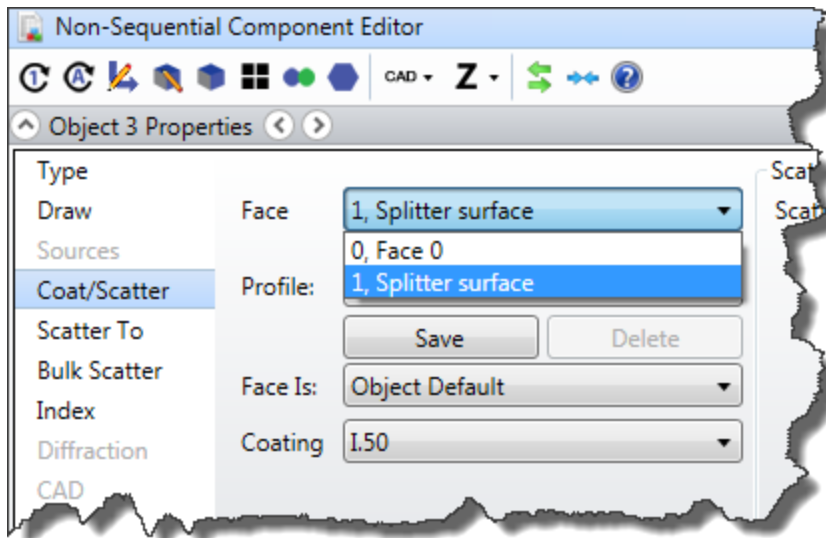
Face 1 is coated with an IDEAL coating I.5, which transmits 50% and reflects 50% of the ray energy, and face 0 is coated with I.95 which transmits 95% and reflects 5% of energy.

All native OpticStudio objects use *faces* to define the various regions of optical interest in the object, and these are documented in the object definition section of the manual. Alternatively, you can use the NSC Object Viewer in the Analyze tab to view individual objects.
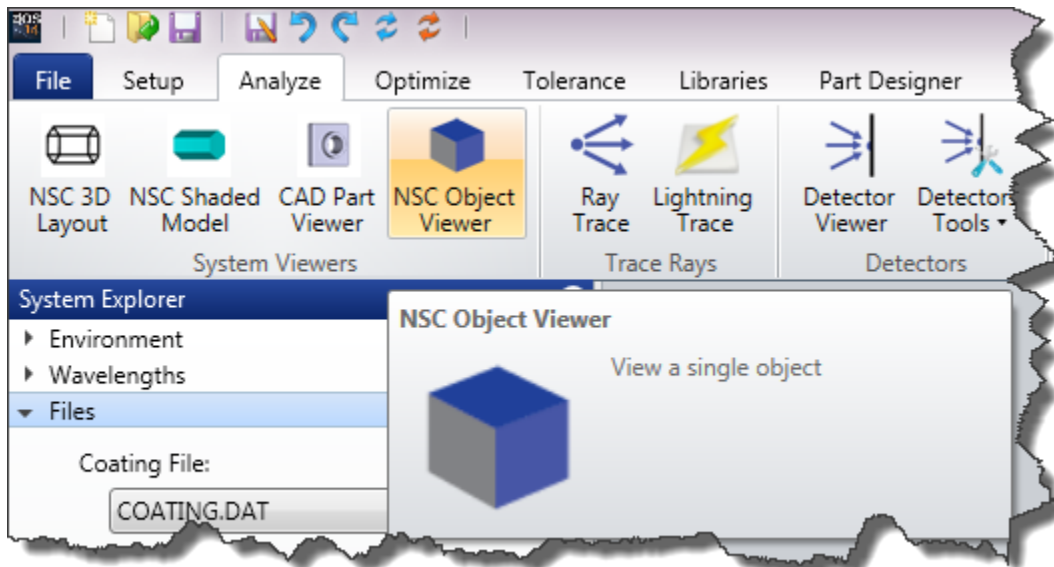
If you click on a face of the object with the mouse, that face will highlight and will be identified in the title bar of the window:
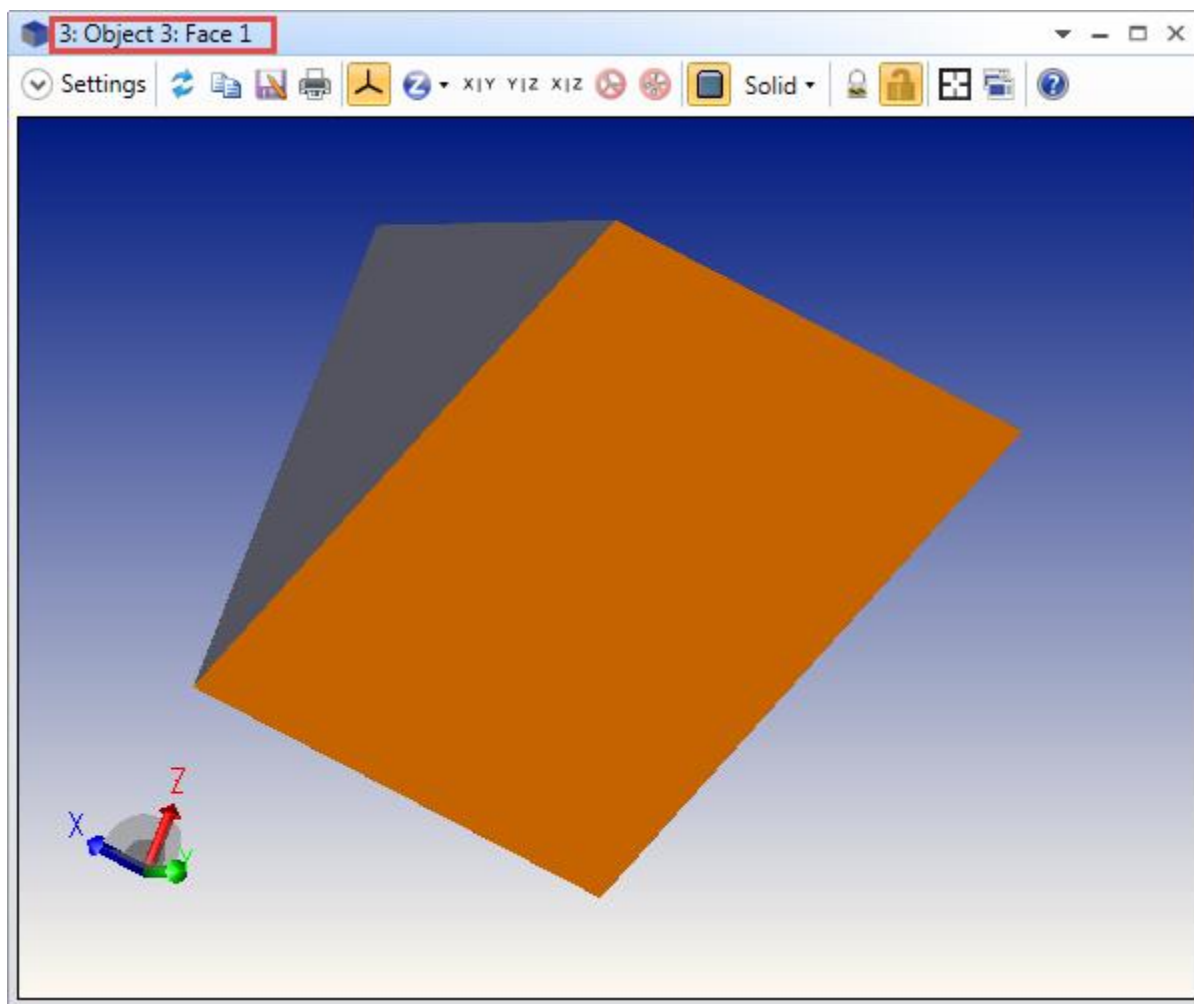
Defining the Faces of CAD objects

OpticStudio uses the several different CAD objects to load STL, STEP, IGES and SAT format files, and also to work dynamically with SolidWorks™, AutoDesk Inventor™ and Creo Parametric™ (formerly Pro/Engineer) parts and assemblies. These are defined by a potentially huge number of NURBS surfaces. Some CAD programs create data files that have many more small surfaces than is useful for optical analysis.

For example, a simple cylinder may be described in the CAD file by hundreds of small surfaces, while for optical analysis only two or three different optical properties are applied to the entire object. Rather than assign optical properties to each of the many surfaces, it is usually more convenient to group CAD surfaces by assigning a single face number to all surfaces that form a continuous smooth portion of the object.

CAD objects support a 'Face Mode' parameter to simplify the allocation of optical faces to an arbitrary CAD part, as follows:

- Face Mode = 0: All surfaces are assigned face number 0. The entire object will have just one face. This is ideal if a single optical finish, like anodized aluminum, will be applied to the whole object

- Face Mode = 1: All surfaces whose edges meet along a non-zero length curve, and whose normal vectors along the curve of contact are parallel within a user defined angle tolerance are assigned a common face number. The angle tolerance is defined by the Face Angle (parameter 8). This mode allows control over how finely the faces

are numbered. If the Face Angle is set to a large value (such as 180) then all faces that touch will share a common number. Larger Face Angles yield fewer unique faces. This is useful with faces that are at say 90° have different optical properties.

- Face Mode = 2: All surfaces are uniquely numbered. This mode yields the largest number of unique faces.

- Face Mode = 3: Retains the face numbers defined in the imported file. Some CAD files, such as those created by Zemax, have face numbers already defined. If Zemax recognizes the face numbers, they will be used. If Zemax does not detect the face numbers, the surfaces will be numbered as for Face Mode = 2.

- Face Mode = 4: All surfaces on each separate object defined in the CAD file are assigned a common face number. This option is useful for applying one property to all surfaces on each object when more than one object is defined with a single CAD file.

OpticStudio also supports manual assignment of face numbers to imported object surfaces using the "CAD" section of the Object Properties.

Tutorial 9.3: Ray Splitting

Either sequential or non-sequential ray-tracing may be done while accounting for polarization effects, or polarization may be ignored. If polarization ray-tracing is being used, transmission, reflection, and absorption of optical energy is accounted for at all surfaces, and bulk absorption by the optical media is also accounted for.

In pure non-sequential ray-tracing, rays may also be split at interfaces. In this case, reflection losses are not just accounted for, but a new ray is launched that takes the reflected energy away.

Because accurate reflection and transmission computation requires polarization information, ray splitting is only allowed when performing polarization ray-tracing.

3D Layout

Beam splitter using coatings
3/5/2014

Zemax
OpticStudio

Beam splitter.ZMX
Configuration 1 of 1
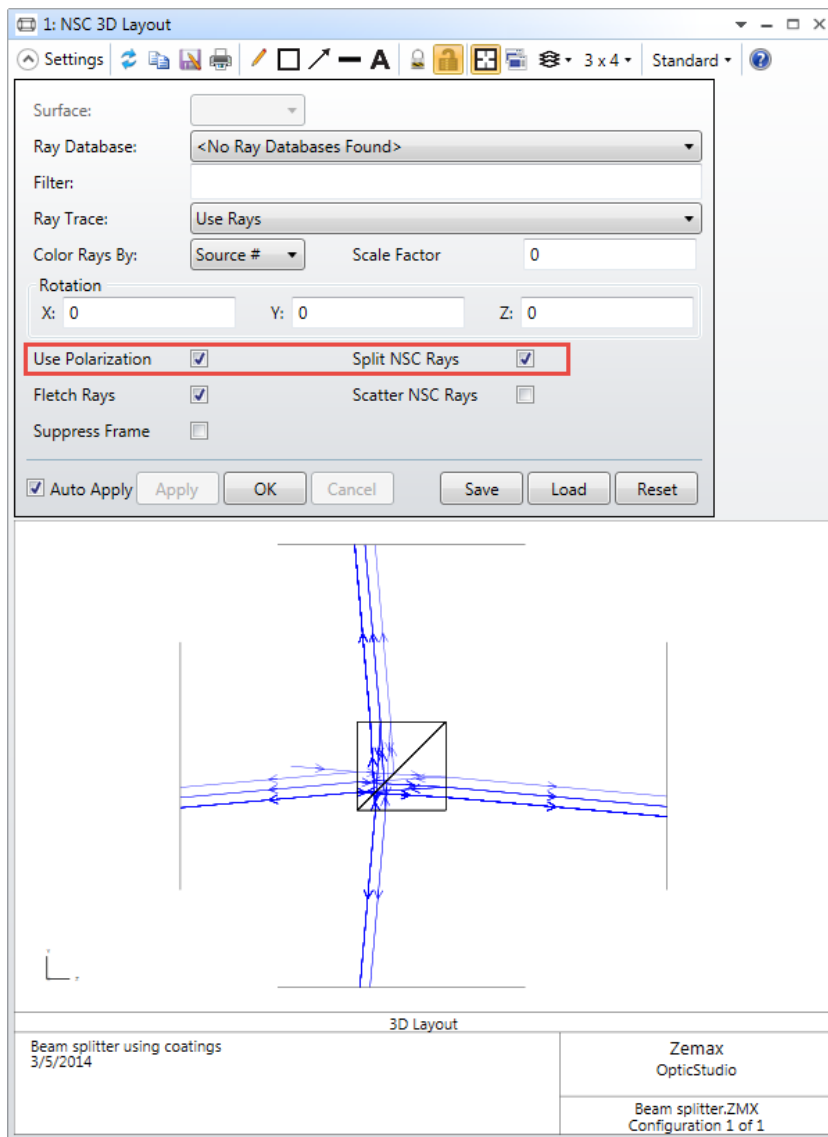
Ray splitting can be turned off, and in this case the transmitted path is always taken at a refractive interface unless the ray totally internally reflects. The reflected path is of course always taken if the object is a mirror.

The layout above shows some of the ray paths possible in the beamsplitter example when rays are split. There is only 1 input ray drawn! The ray termination criteria defined in the System Explorer's Non-Sequential section are essential for efficient calculation when ray splitting is on. It is advisable to set the relative ray transmission reasonably high, around 0.001, until the model is working well and more detailed results are needed.
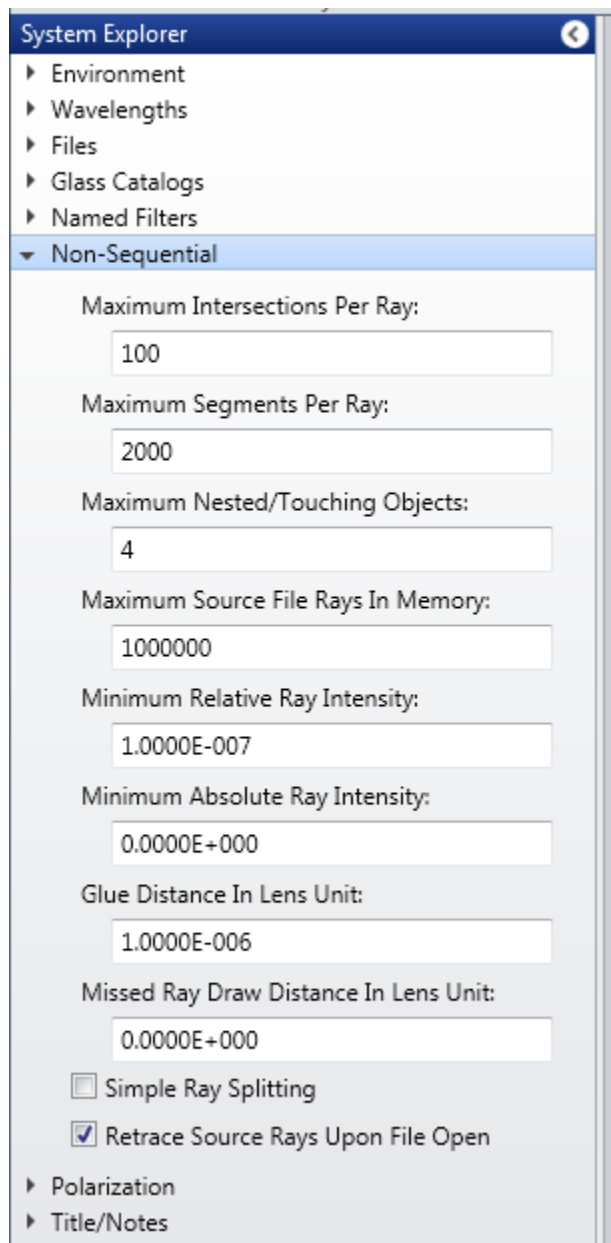
OpticStudio also supports an option to randomly choose either the reflected or the refracted path rather than split the ray into two and trace both. This is controlled by the 'Simple Ray Splitting' shown above. The decision to trace the reflected or the refracted ray is random; with the reflection and transmission coefficients being interpreted as a relative probability of taking that path.

Tutorial 9.4: Ray Scattering

In addition to partial reflections at the surfaces of optical components, rays can also scatter due to microscopic roughness of the surface. OpticStudio supports many detailed models of scattering from optical surfaces, including Lambertian (used for very rough, highly scattering surfaces), Gaussian (typically used for modeling the scattering of a well-polished surface, ABg, K-correlation and more. In addition, OpticStudio can import scattering data in a simple ASCII file format,

and the Premium edition contains a library of measured scatter profiles. See the Scattering group in the Libraries tab for details
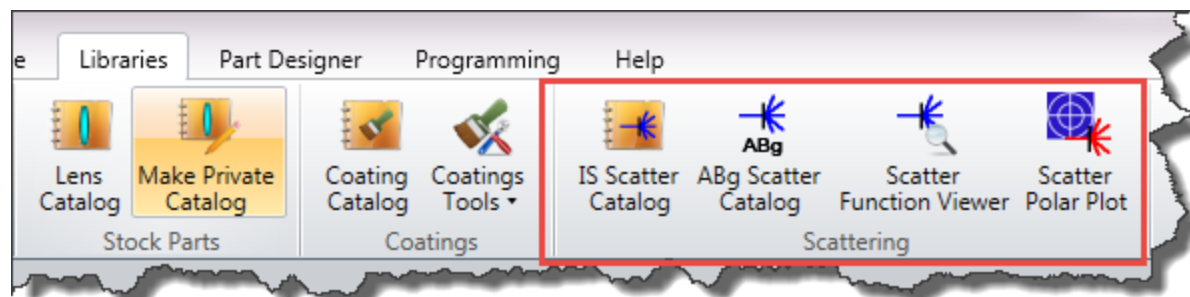
Although scattering can be used in sequential ray-tracing (see the Scattering tab Properties Inspector dialog), it is most useful in non-sequential ray-tracing, where rays can go wherever they want to. Scattering functions are applied to the faces of non-sequential objects in the same way as thin-film coatings are: on the Coat/Scatter tab of the Object Properties tab.

Open the sample file in Non-sequential\Scattering called ABg scattering surface.zmx. This file uses the ABg scattering model on Object 2, which is commonly used with measured scattering data:
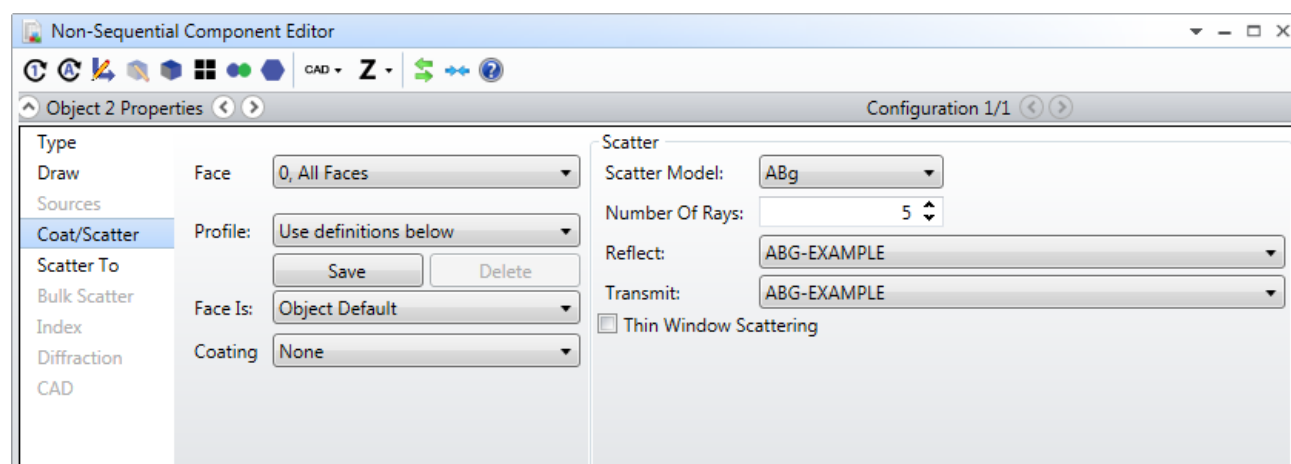
As it is based on measured data, the total amount of energy scattered by the surface is defined by the data file. Other scattering models, like the Lambertian model, require you to tell OpticStudio how much energy to scatter using the "Scatter Fraction" parameter on this dialog, which will appear if a scattering mode that requires it is chosen:
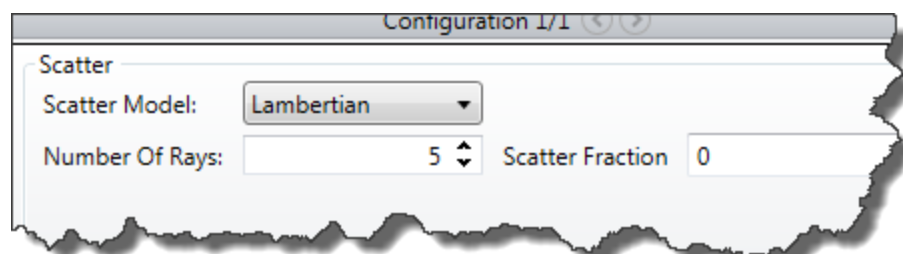
If ray splitting is turned off, the ray will either scatter or not, depending on the value of the 'Scatter Fraction' parameter (or equivalent measured data) and a random number OpticStudio generates for each ray-object intersection. Update the NSC 3D layout, and note that *the ray either scatters or does not scatter*:

3D Layout

Scattering surface using ABg
3/5/2014

Zemax
OpticStudio

ABg scattering surface.ZMX
Configuration 1 of 1
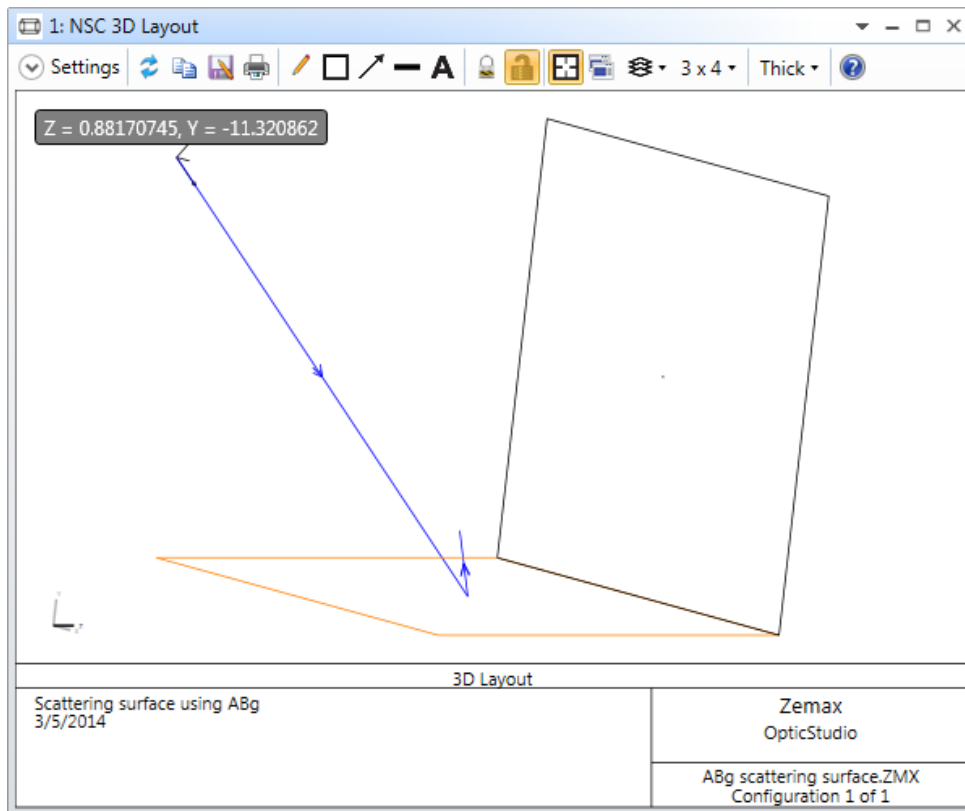
On the Settings dialog, select "Split Rays' as well as "Scatter Rays":
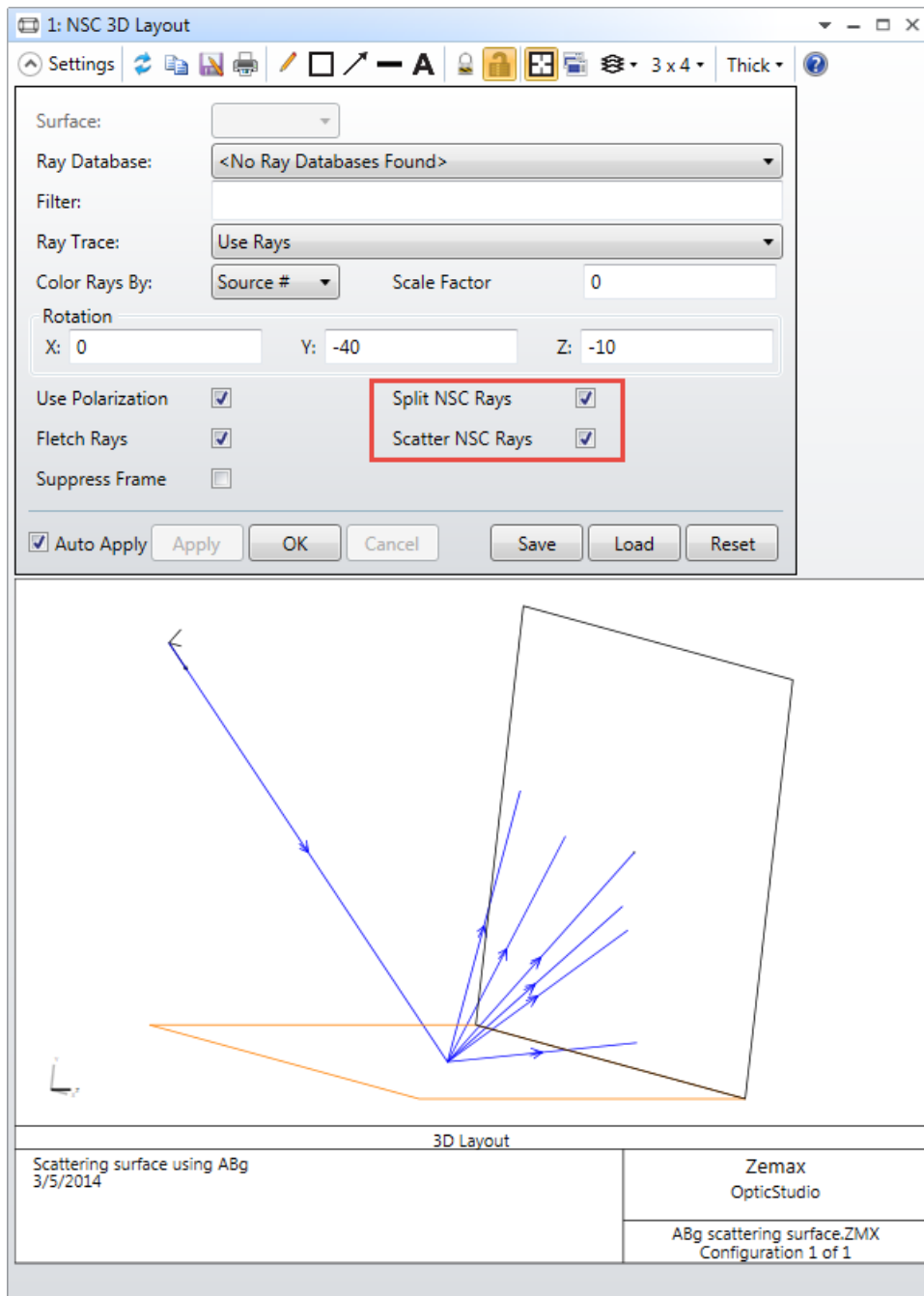
The ray will now be split into the unscattered ray, and five scattered rays.

Update this layout a few times, and note that you always get the unscattered ray, and five randomly scattered rays. The number of scattered rays is defined by the 'Number of Rays' parameter in the Coat/Scatter tab of the Object properties.

Tutorial 9.5: Importance Sampling

A very large number of rays may need to be traced to find a relatively small number of scattered rays that strike an object of interest, such as a detector. OpticStudio supports two powerful ways to improve the efficiency of the scattering analysis.

The first method is to scatter a ray according to the scatter distribution, but only trace the ray if the ray propagates towards an object of interest. This method may be implemented by defining a 'Scatter To' list of objects using the 'Scatter To' tab in the Property Inspector.
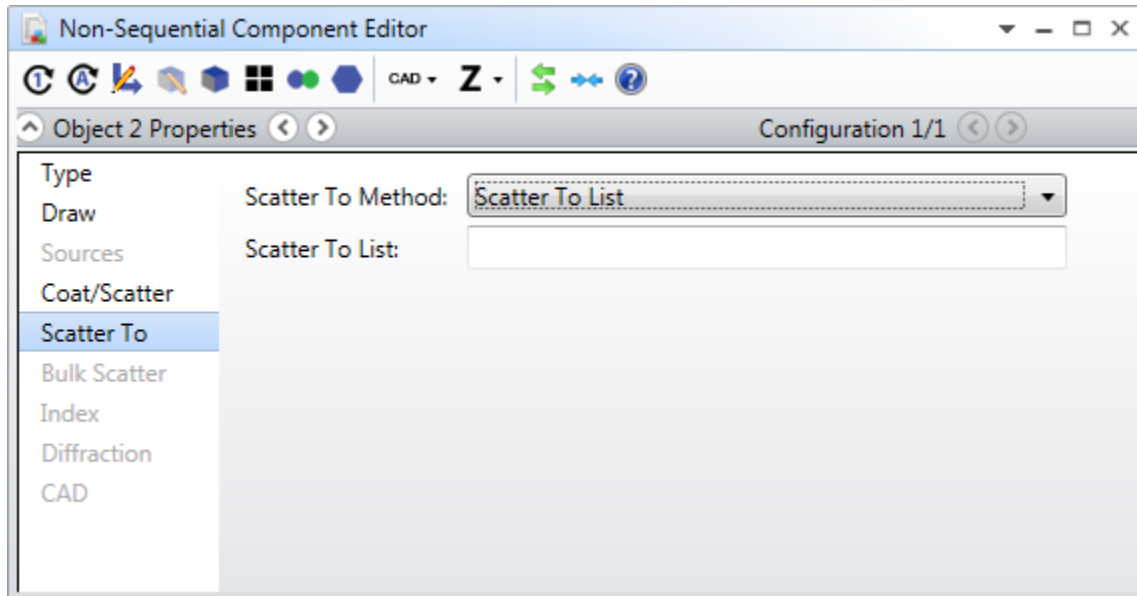
The Scatter To method works well for wide angle scatter (such as Lambertian scatter) and when the object of interest subtends a relatively large angle as seen from the scattering surface.

The second method is to always scatter the ray towards the object(s) of interest, and then to normalize the energy the ray carries to account for the probability the ray would have actually scattered in that direction. This method is called "Importance Sampling". Importance Sampling is generally superior to the Scatter To method if the scatter is narrow angle or the object of interest subtends a relatively small angle as seen from the scattering surface.

Open the sample file in Non-sequential\Scattering called Importance Sampling Demonstration.zmx. This shows directly the benefit of using Importance Sampling. Rays are always scattered to the desired object, in this case a detector, and the resulting signal/noise ratio is therefore far superior.
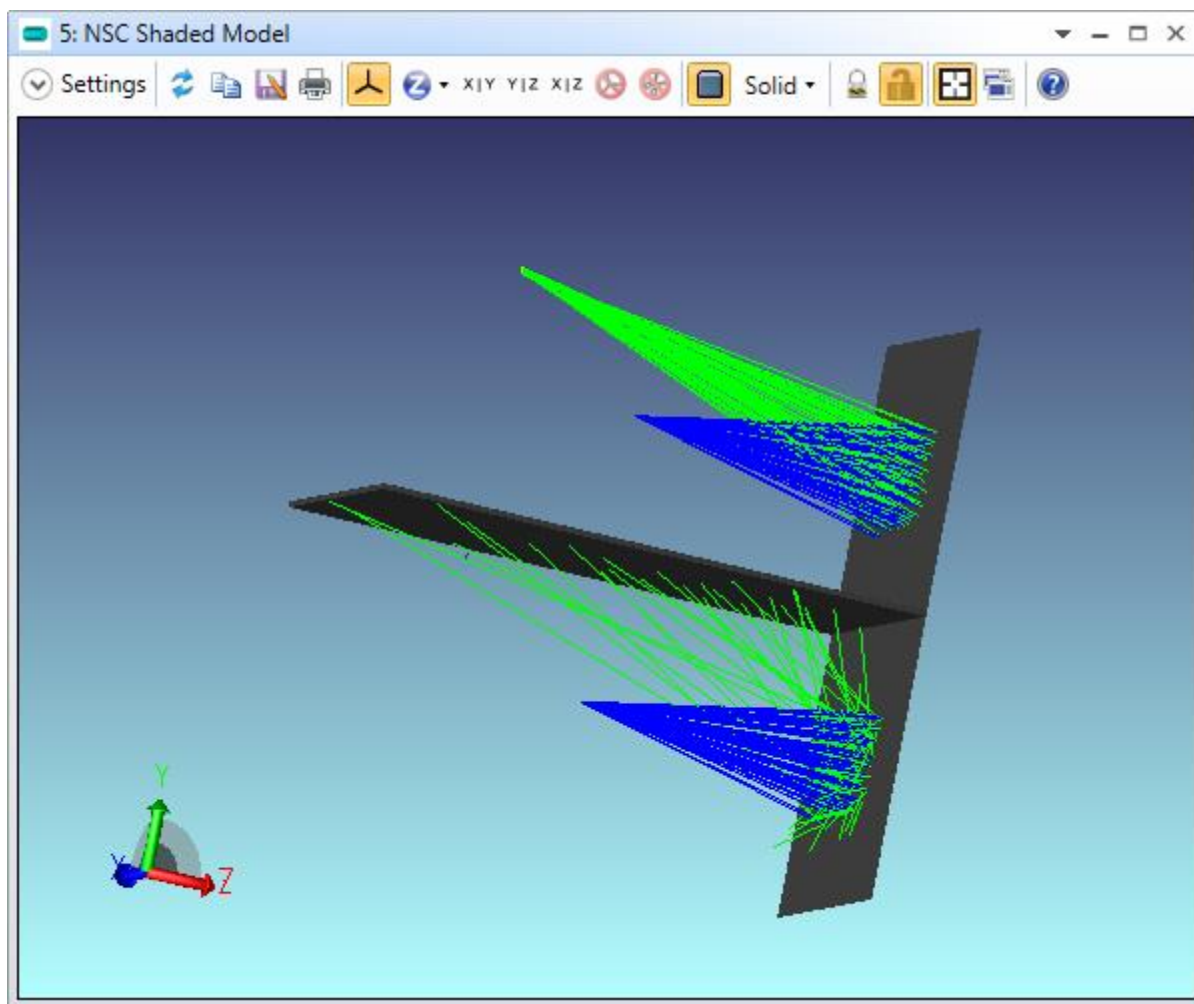
Tutorial 9.6: Bulk and Fluorescent Scattering

Bulk scattering models the random scattering of rays while propagating through a solid object. This may be a very rare event (like scattering from inclusions in optical-quality glass) or a very common event (like scattering in biological tissue samples). OpticStudio includes several bulk scattering models, including Henyey-Greenstein and Rayleigh scattering.

In addition, rays may change wavelength when bulk scattered, usually to longer wavelengths. Open the sample file in Non-sequential\Scattering called Fluorescence Example.zmx. This file uses two wavelengths, 1 (blue) and 2 (red). The source radiates only in the blue, and this light enters a medium that scatters the light in angle and wavelength. A beamsplitter transmits the red and reflects the blue:
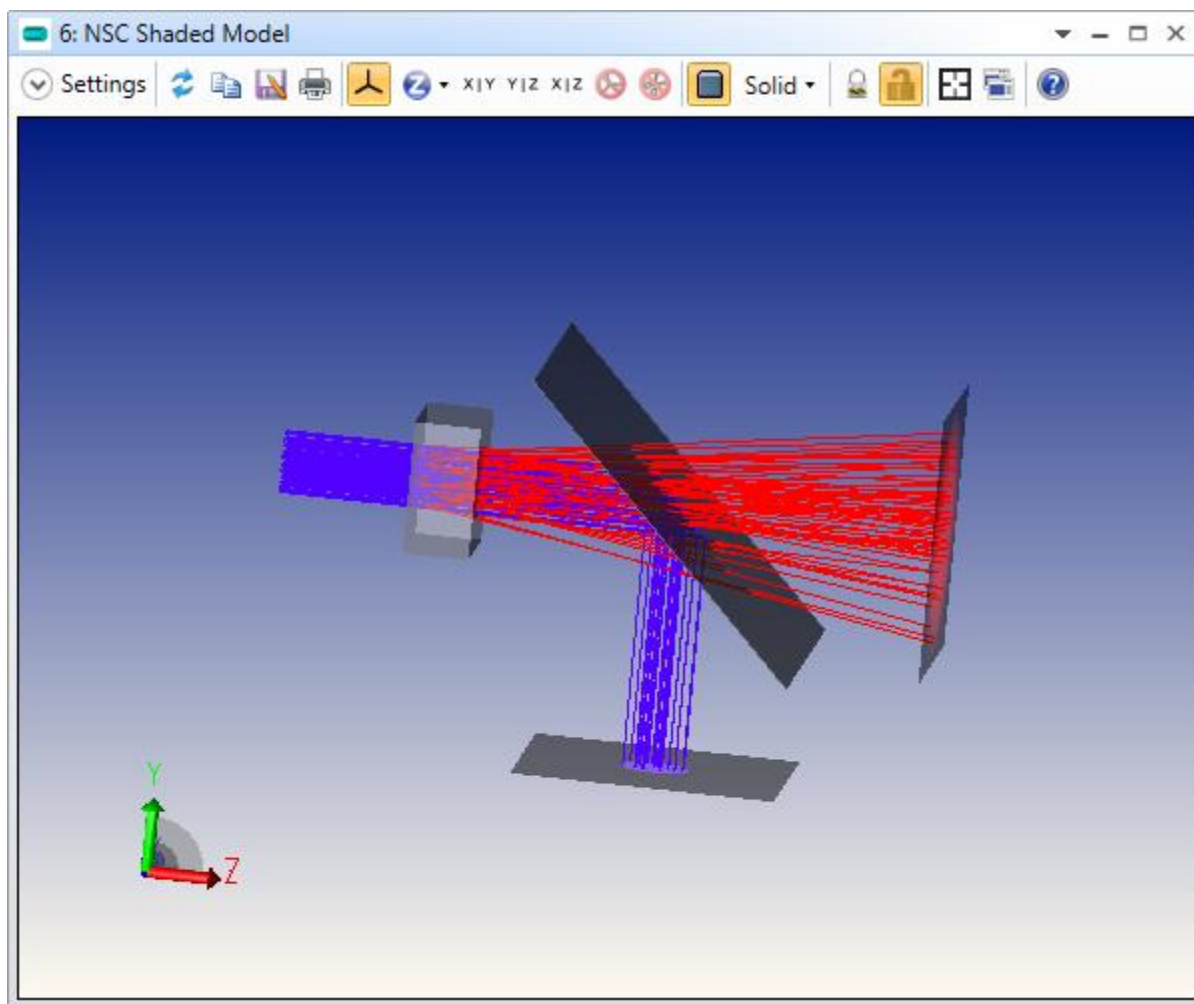
Bulk scattering properties are defined on the 'Bulk Scatter' tab of the Property Inspector dialog:
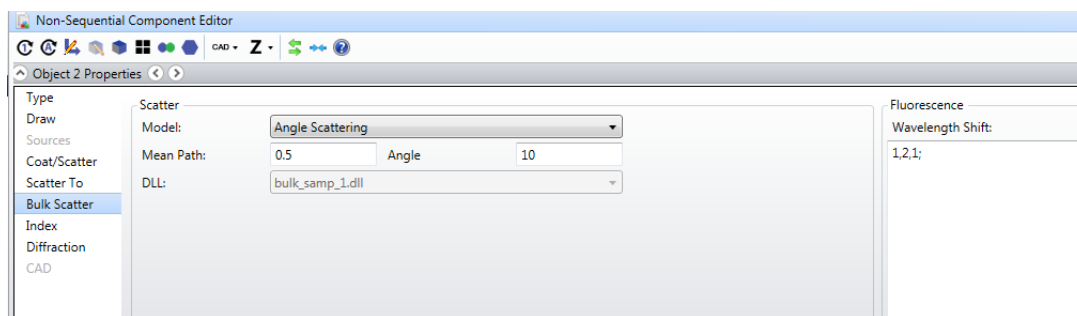
The "Wavelength Shift" control allows definition of wavelength transitions during bulk scatter events. The syntax is "in, out, prob" where "in" is the input wavelength number, "out" is the output wavelength number, "prob" is the relative probability that this shift will occur when tracing the in wavelength. Multiple transitions may be defined using a semi colon separator.

For example, if a single input wavelength (#1) will shift to wavelength #2 50% of the time, wavelength #3 40% of the time, and the remaining 10% of the time will remain at the input wavelength, then the wavelength shift string is

```
1, 2, 50.0; 1, 3, 40.0; 1, 1, 10.0
```

What's Next?

Well done on working through this guide! This guide is by necessity brief, and has only scratched the surface of what you can achieve with OpticStudio.

The goal was to get you started using OpticStudio in just an hour or two. There is plenty that is not covered. For example, it does not discuss Tolerancing, Physical Optics, thermal analysis, multi-beam interference in non-sequential mode, and many of the powerful analysis and optimization capabilities of OpticStudio.

Note: Just because it is not covered here does not mean that it's not in OpticStudio!

Here are some other resources that will help you:

- The OpticStudio Knowledge Base. This is an indispensable resource for OpticStudio users, and contains hundreds of articles including tutorials, answers to frequently-asked questions, and examples. It is structured into categories, and has a powerful search engine to help you find articles of interest easily.
- The Technical Reference files in the Help system give detailed discussions of the underlying capabilities of OpticStudio
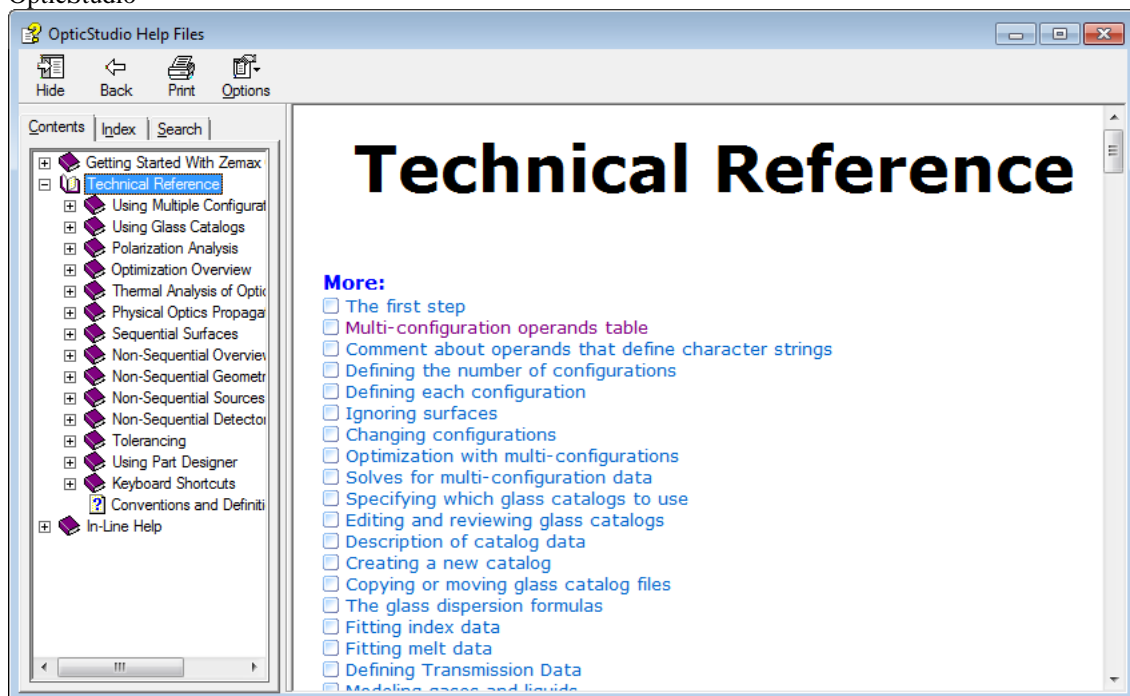
- The In-Line Help files are accessed by clicking the Help "?" button in dialog boxes, and give detailed descriptions of the different analysis features.
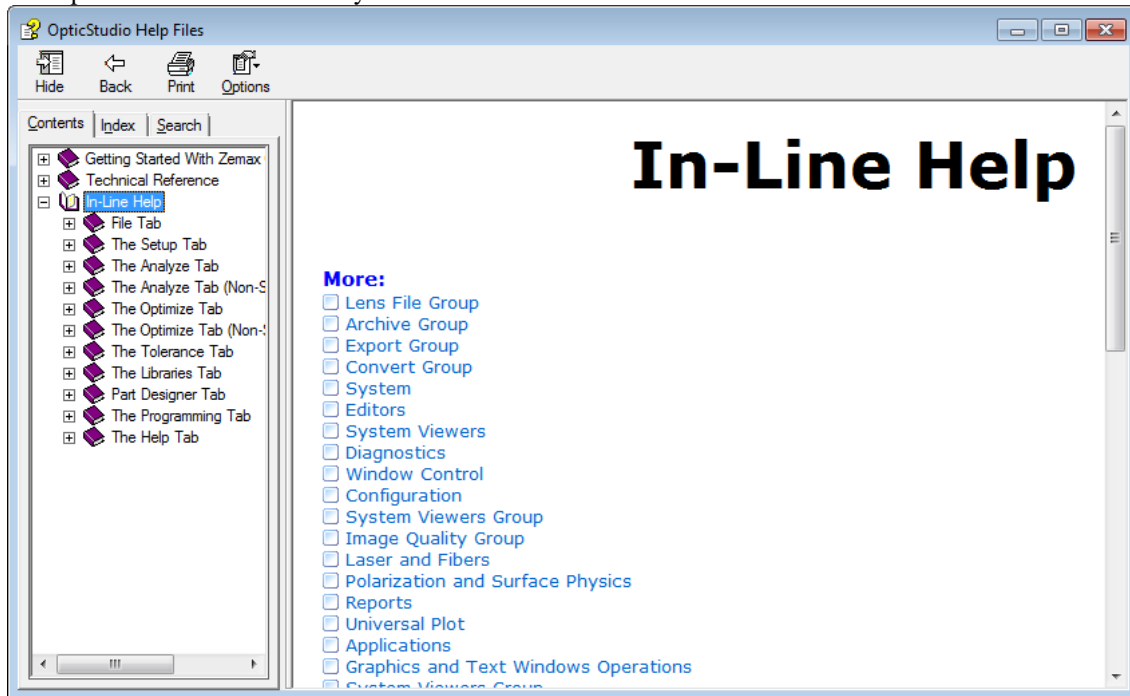
- Zemax LLC and its team of international distributors offer dedicated training classes on the use of OpticStudio. Courses are held from introductory to advanced level, and cover all aspects of sequential lens design, illumination, stray light, programming OpticStudio and more. See www.zemax.com/training for more details.

Getting Technical Support

Please contact support@zemax.com, or your local distributor, if you need help in using OpticStudio! For help in the Chinese language, use china@zemax.com.

Note: It is vastly easier to resolve a technical problem with a sample file that demonstrates it. Of course, many customers are concerned about the confidentiality of their designs. We do not undertake any product development contract work. Therefore, if you have a technical question, you can be sure that you are not explaining your work to someone who may be working for a competitor, or bidding against you for the same job! Our goal is simply to help you use OpticStudio most effectively.

If you are still uncomfortable about sharing your design file, try simplifying it to the bare essentials necessary to show the problem, or use one of the >100 sample files supplied with OpticStudio instead.

Then please follow these steps:
- Make sure you are running the current version. Check For Updates in the Help tab. If you are eligible for technical support then you are eligible to run the current version. Make sure you can reproduce your problem with the current version as it contains fixes for all known bugs.
- Run System Check in the Setup tab to test your file for common setup errors.
- Email a full description of the problem to support@zemax.com or your local distributor. Please be sure you include:
  - Your name, organization, email address and phone number in case we need to talk directly with you.
  - The serial number of your OpticStudio key (run Help...About in OpticStudio to get this).

- A clear description of your problem. Include the sample file that demonstrates the issue if you can.
- Use File…Backup to Archive and send the resulting .ZAR file. This single file contains everything we need to reproduce your design exactly as you have it on your computer.

You will get a response within a maximum of one working day, and usually a lot sooner.

If support for your key is not current, you can renew it easily by sending an email to sales@zemax.com or your local distributor and requesting a quotation for support renewal.

References on Lens Design

Neither the Zemax program nor the Zemax documentation will teach you how to design lenses or optical systems. Although the program will do many things to assist you in designing and analyzing optical systems, you are still the designer. The Zemax documentation is not a tutorial on optical design, terminology, or methodology. Technical support available to Zemax users includes assistance in using the program, but does not include tutoring on fundamental optical design principles. If you have little or no experience in optical design, you may want to read up on any of the many good books available on the subject. The following table lists some (but by no means all) of the books which will aid in your education.

REFERENCES

Author	Title	Publisher
Bass	Handbook of Optics	McGraw-Hill
Born & Wolf	Principles of Optics	Pergamon Press
Fischer & Tadic-Galeb	Optical System Design	McGraw-Hill
Geary, Joseph M.	Introduction to Lens Design: With Practical Zemax Examples	Willmann-Bell
Hecht	Optics	Addison Wesley
Kingslake, Rudolph	Lens Design Fundamentals	Academic Press
Laikin, Milton	Lens Design, Third Edition	Marcel Dekker
Mahajan, Virendra	Aberration Theory Made Simple	SPIE Optical Engineering Press
O' Shea, Donald	Elements of Modern Optical Design	John Wiley and Sons
Rutten and van Venrooij	Telescope Optics	Willmann-Bell
Shannon, Robert	The Art and Science of OpticalDesign	Cambridge University Press
Smith, Gregory Hallock	Practical Computer-Aided Lens Design	Willmann-Bell, Inc.
Smith, Warren	Modern Optical Engineering	McGraw-Hill
Smith, Warren	Modern Lens Design	McGraw-Hill
Welford	Aberrations of Optical Systems	Adam Hilger Ltd.

| Welford | Useful Optics | University of Chicago Press |

Most importantly, Zemax is not a substitute for good engineering practices. No design should ever be considered finished until a qualified engineer has checked the calculations performed by the software to see if the results are reasonable. This is particularly important when a design is to be fabricated and significant costs are involved. It is the engineers responsibility to check the results of Zemax, not the other way around.

Converting from Zemax13 and Older

This section of the Getting Started Using Zemax OpticStudio guide is aimed at those users upgrading from earlier versions of Zemax (Zemax 13 and earlier). In this section we map the user interface of the previous versions of Zemax to their current locations, and highlight any changes of significance.

We also recommend you read the Navigating the OpticStudio Interface section of this help file for full instructions on how to use the new UI.
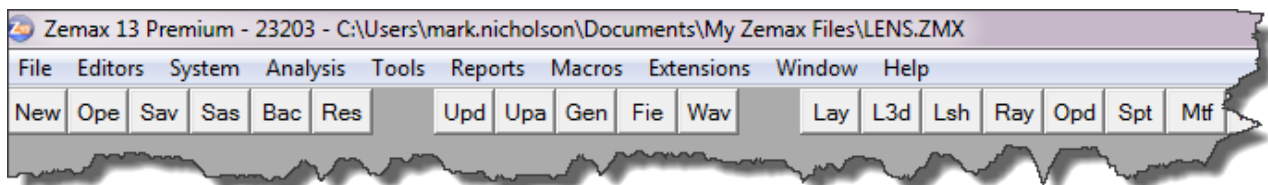
In the following section we will look at each Z13 menu and see where its contents have now moved to.

Note that the button bar used in Z13 is obsolete, and instead you place any feature you want directly onto the Quick-Access toolbar in OpticStudio:
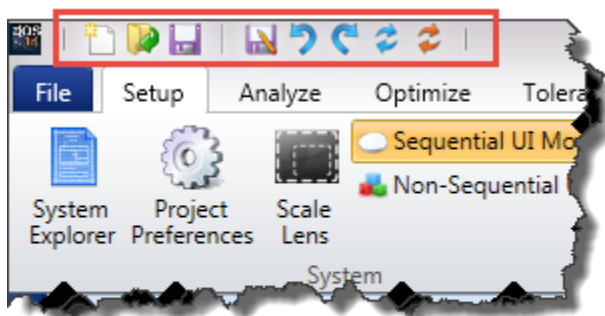
In the Setup tab, click on the Project Preferences icon and simply drag and drop the features you want onto the Quick-Access toolbar.
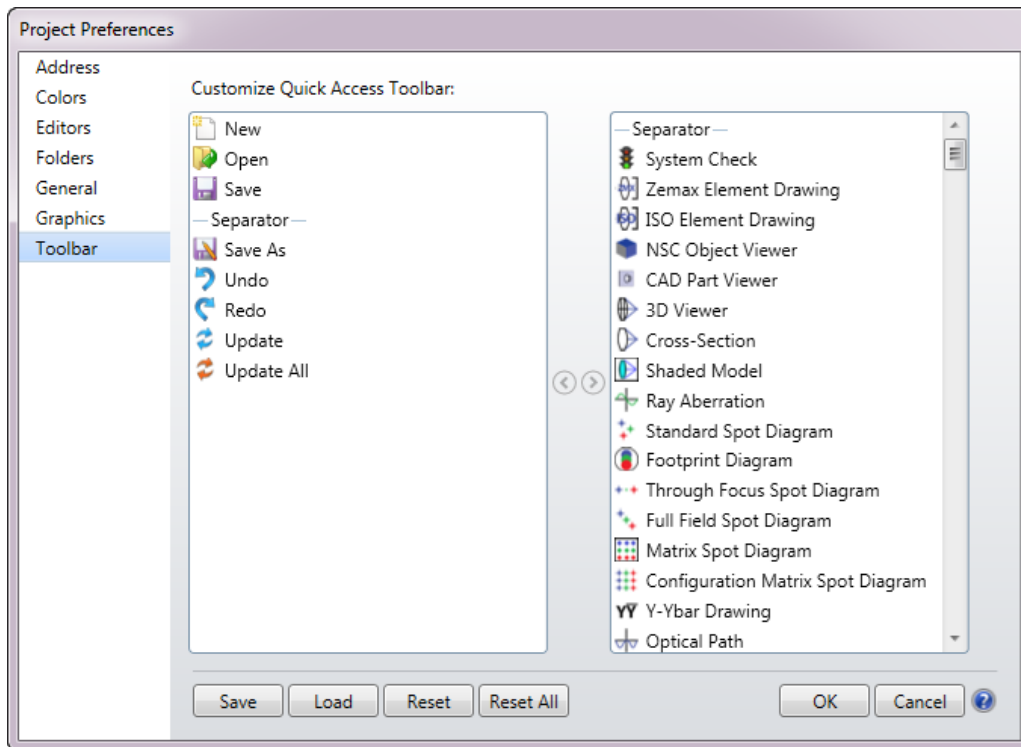
All keyboard shortcuts used by Z13 still work in OpticStudio!

Zemax 13 File Menu
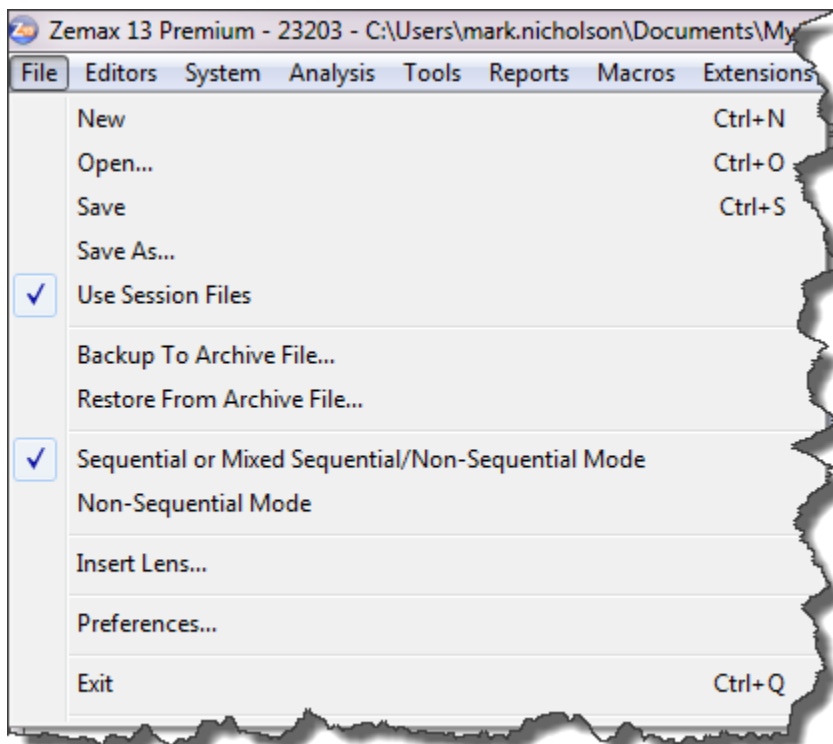
Almost all of these features have moved to the File tab in OpticStudio
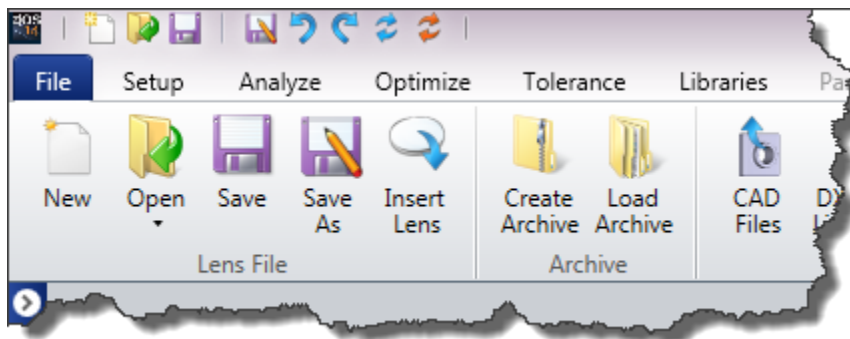
Except for the Program Mode and Preferences, which are now on the Setup tab:
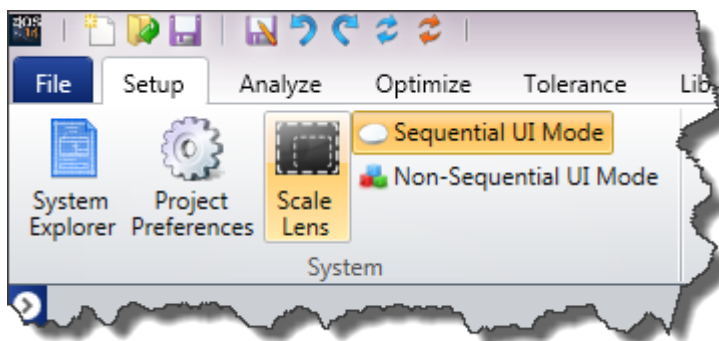
Zemax 13 Editors menu
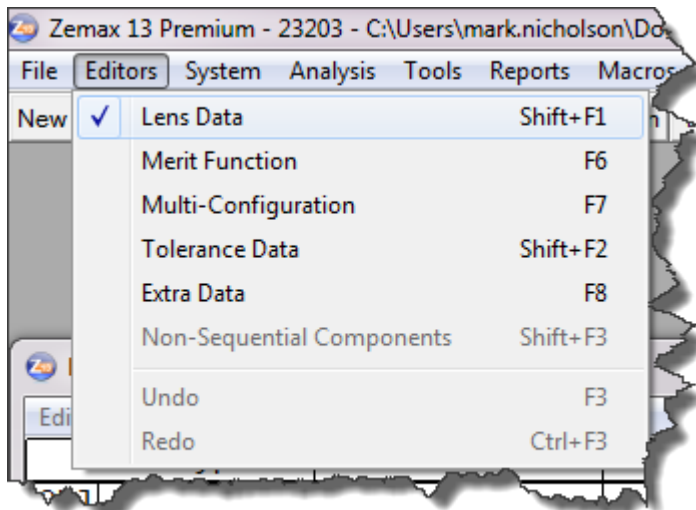
The biggest change here is that the Extra Data Editor is now obsolete. Extra Data is now appended onto the Lens Data Editor. The Lens Data Editor in Z13 uses parameters 0 through 12. In OpticStudio, extra data now appears from Parameter 13 onwards, and ther LDE simply scales in size to accommodate however much data you need.

All Z13 and older files are read and remapped automatically by OpticStudio. Tolerance operands, multiconfiguration operands and optimization operands that referred to Extra Data parameter numbers are now remapped to use parameter data operands instead. As a result, .zmx files produced by OpticStudio can not be read successfully by older versions if the file contains extra data operands in the merit, multi-configuration or tolerance editors.

In Extensions and ZPL macros you can address (previously) Extra Data items using either extra data operands or parameter operands with the parameter number >12.

Editors can be found in the Setup tab in the Editors group, which shows the two main editors and can expand to show all editors:
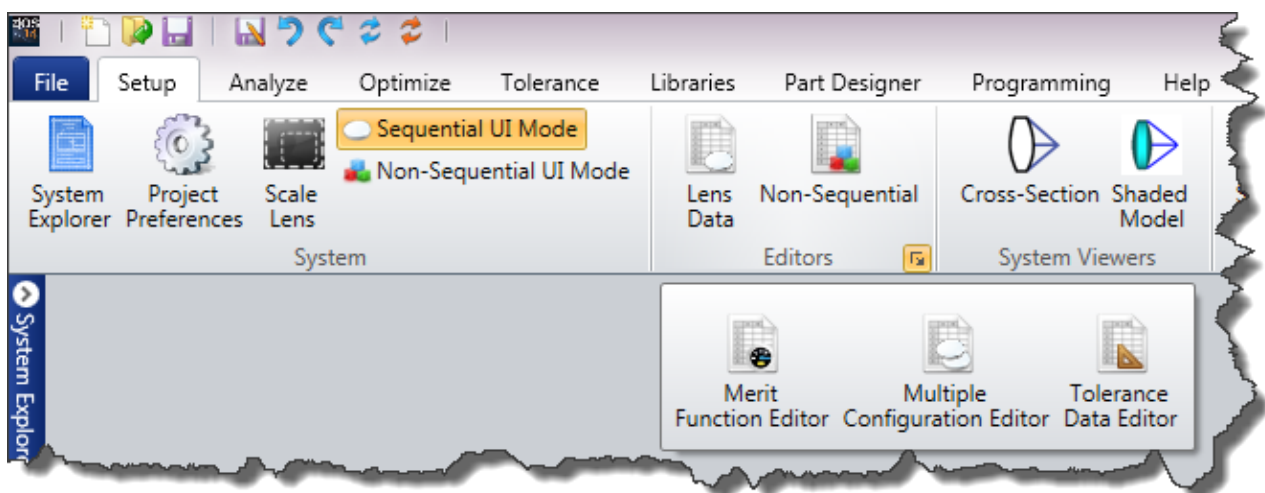
The Multi-Configuration Editor is also available from the Configuration group of the Setup tab:
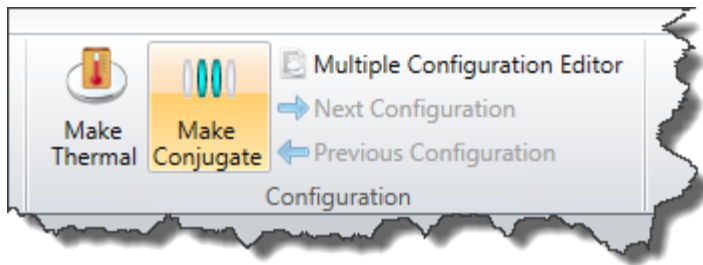
If more than one configuration exists in the MC editor, a smaller Configuration group appears on the Analyze, Optimize and Tolerance tabs also:
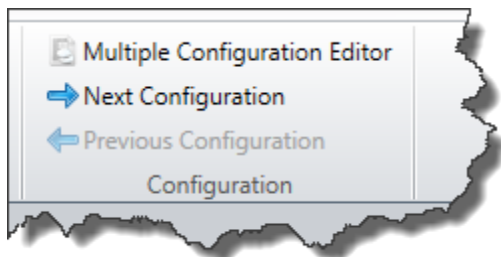
The Merit Function Editor is also available in the Optimize tab, and the Default Merit Function tool is renamed the Optimization Wizard:
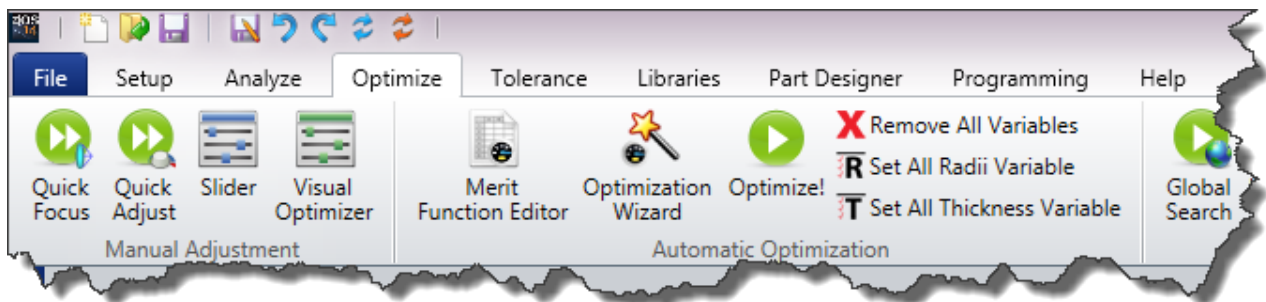
The Tolerance Data Editor is also available in the Tolerance tab, and the Default Tolerances tool is now called the Tolerance Wizard:
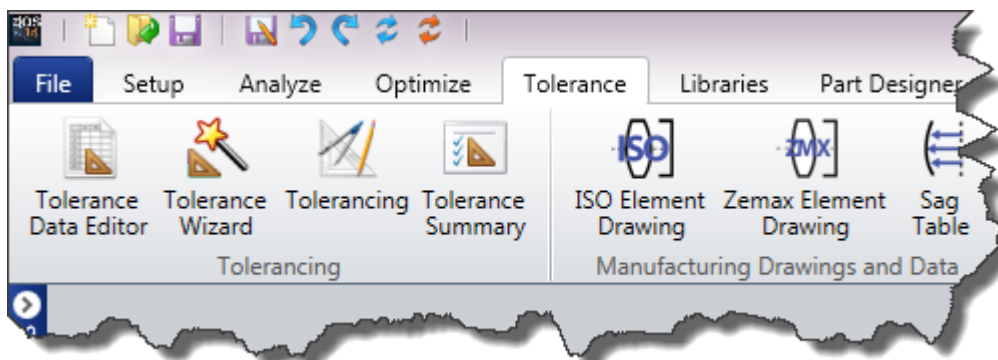
The Editors have been completely re-written to provide a modern, intuitive spreadsheet capability. See the Navigation guide for full details

Zemax 13 System Tab
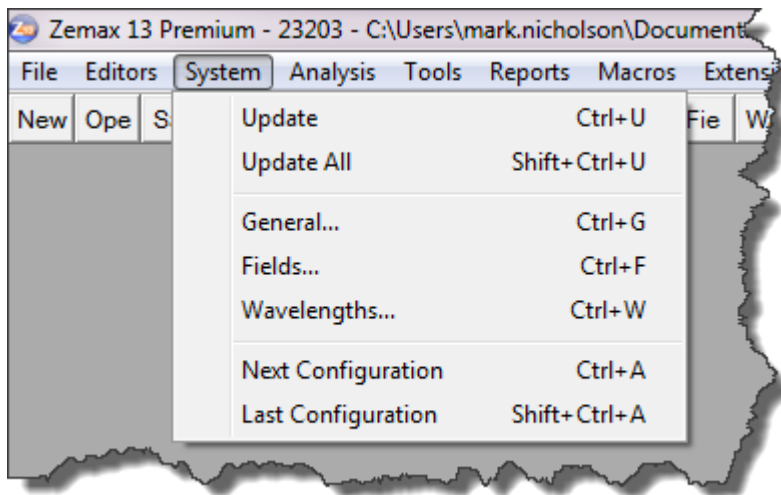
Update and Update All are moved to the Quick-Access toolbar:
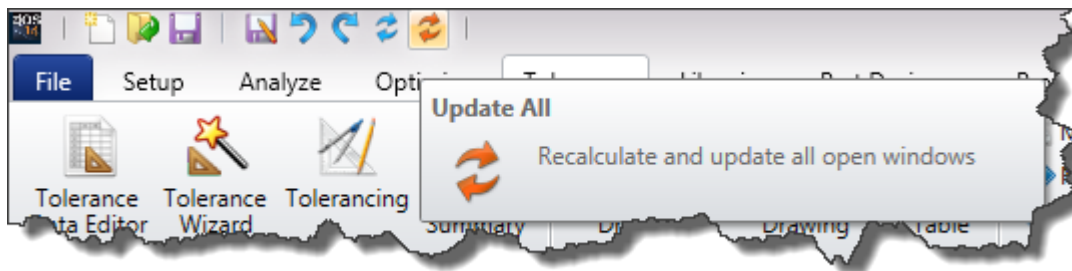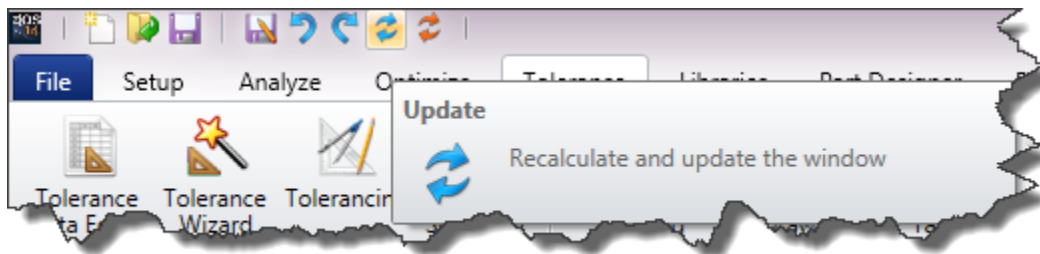
The General, Fields and Wavelengths menu items are now combined in the System Explorer, which is available via the sidebar or the System Explorer icon in the Setup tab:
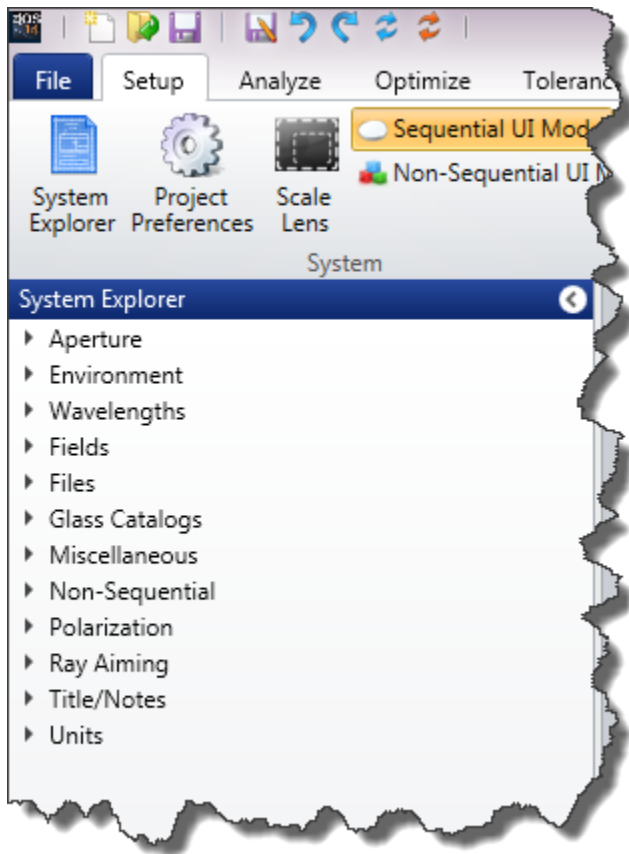
Filed and Wavelength data can be entered directly in the System Editor, or their older-style dialogs can be invoked by double-clicking on the top lines of either the Fields or Wavelengths group:
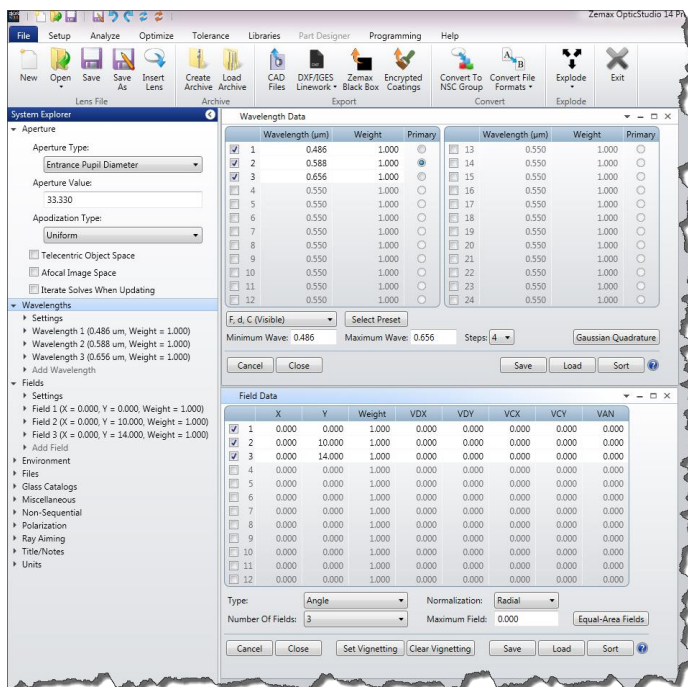
The Next and Last Configuration controls are on the Configuration group of the Setup tab, and if the design has more than one configuration are also available on the Analyze, Optimize and Tolerance tabs.
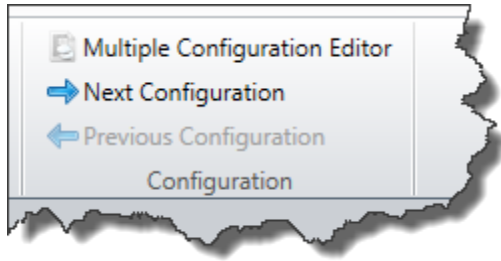
Zemax 13 Analysis Menu
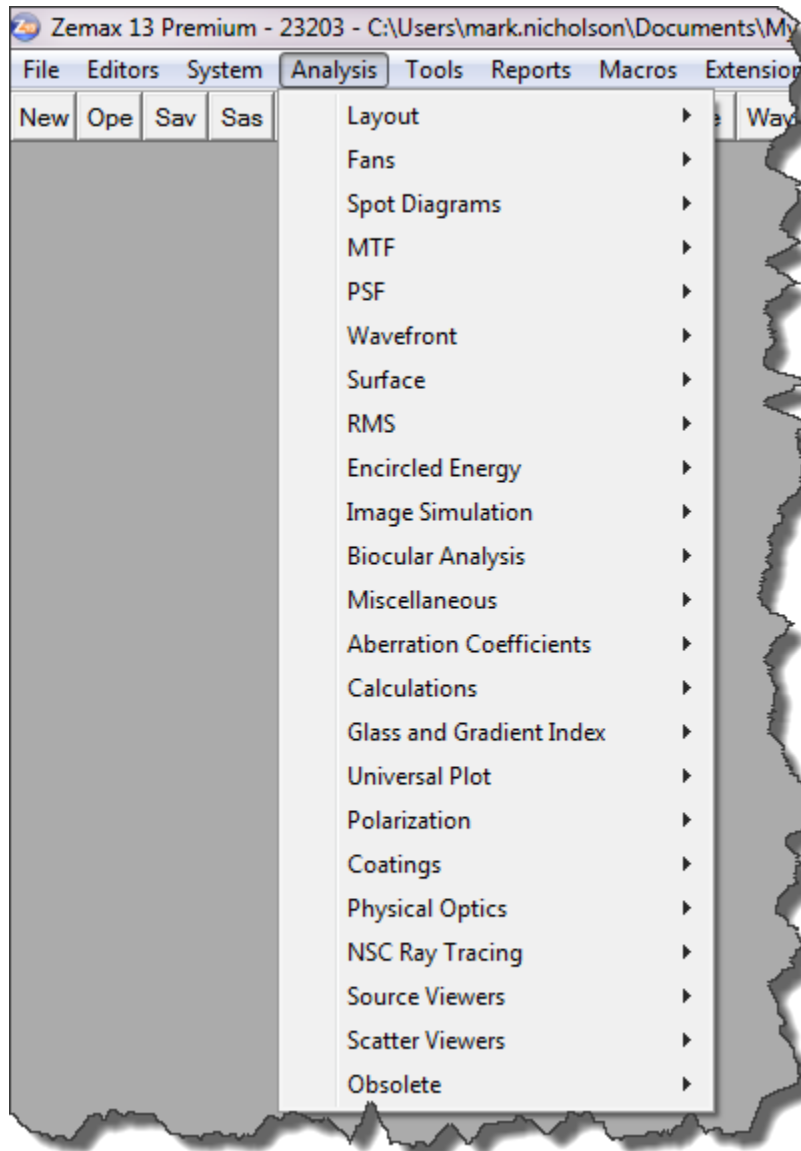
Almost everything on this menu has moved to the Analyze tab:
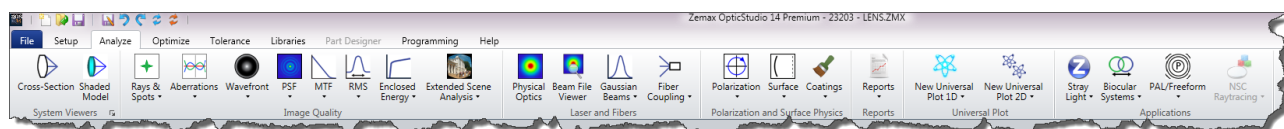
With the following exceptions:

- Glass and Gradient Index now appears in the Optical materials group of the Libraries tab
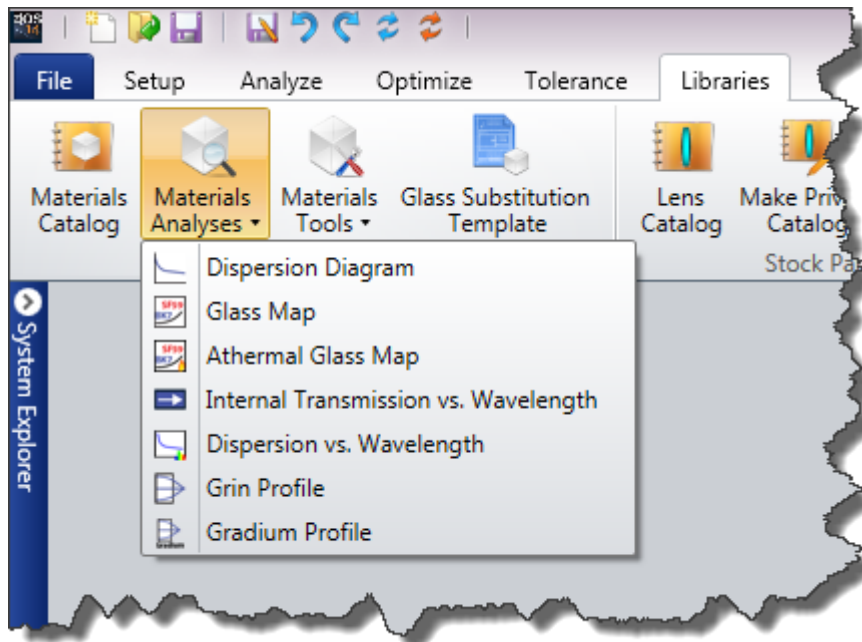
- Source Viewers have moved to the Source Viewers group in the Libraries tab
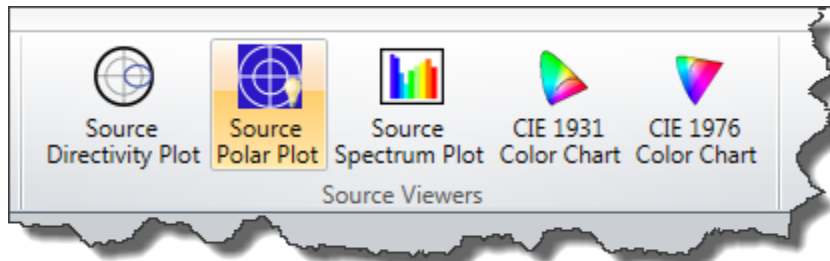
- Scatter Viewers have moved to the Scatter Viewers group in the Libraries tab:
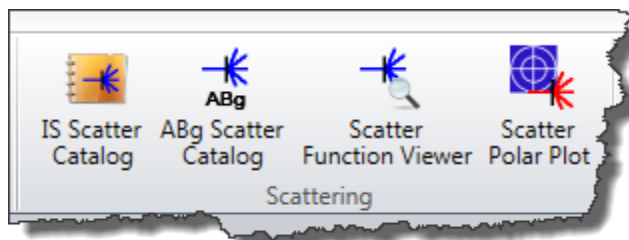
- The items under Miscellaneous have been placed in more meaningful locations:
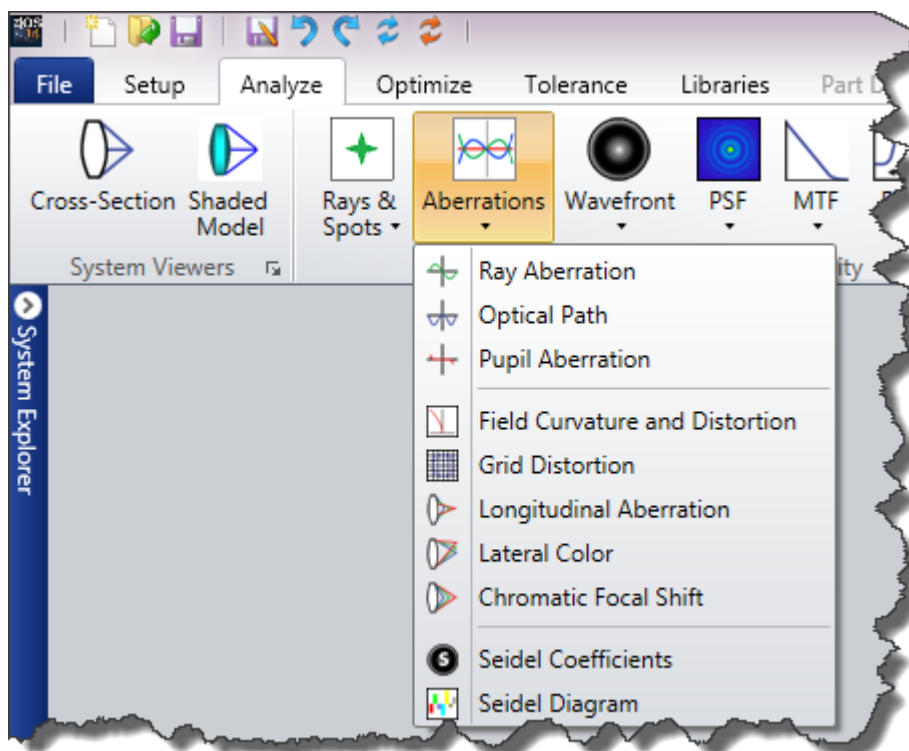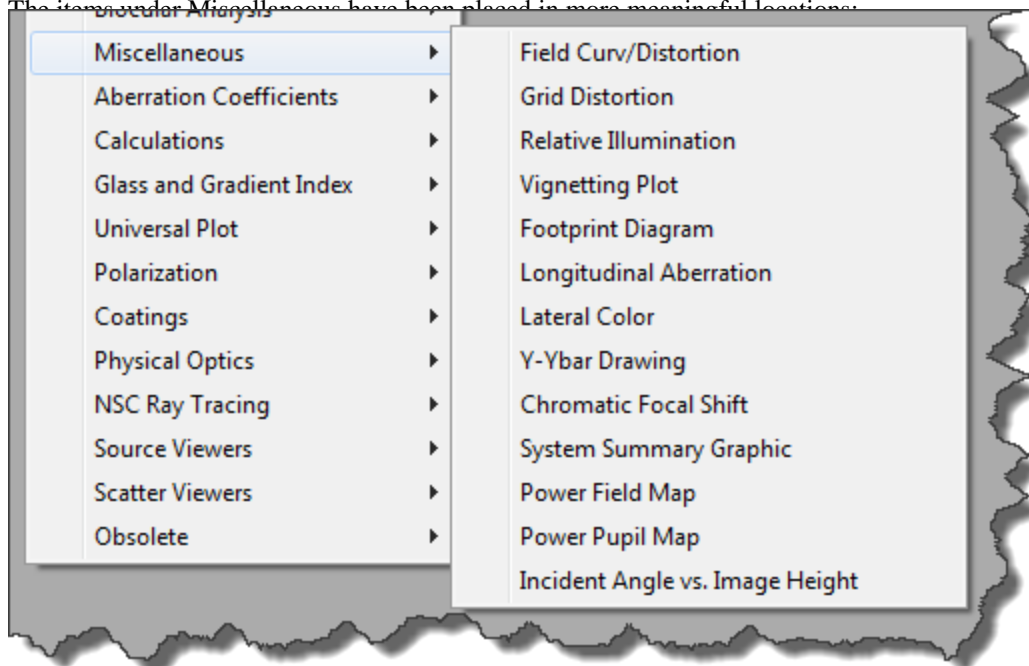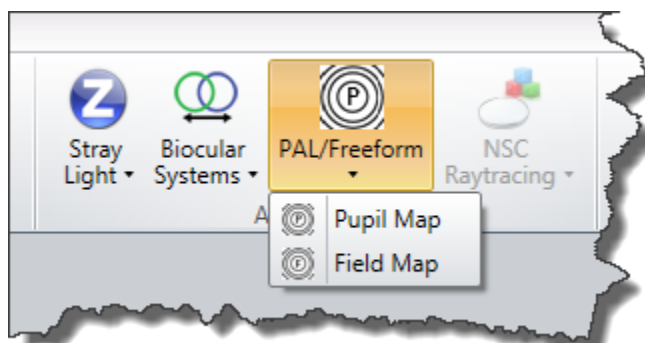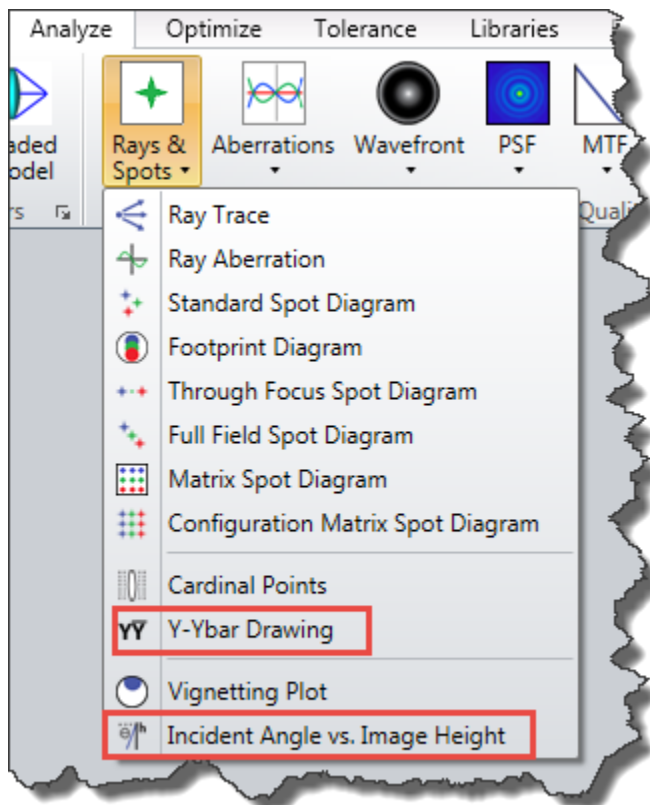
- The items under Obsolete have been removed, except for
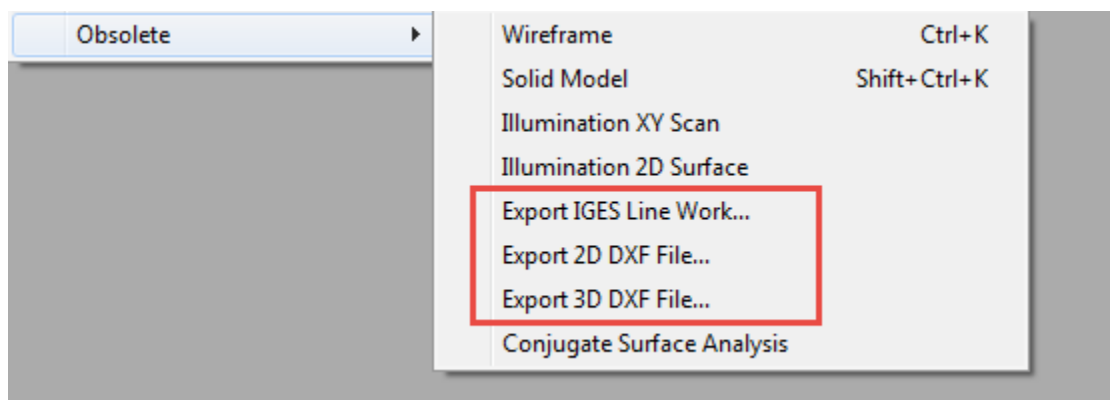
which now appear in the Export group of the File tab:
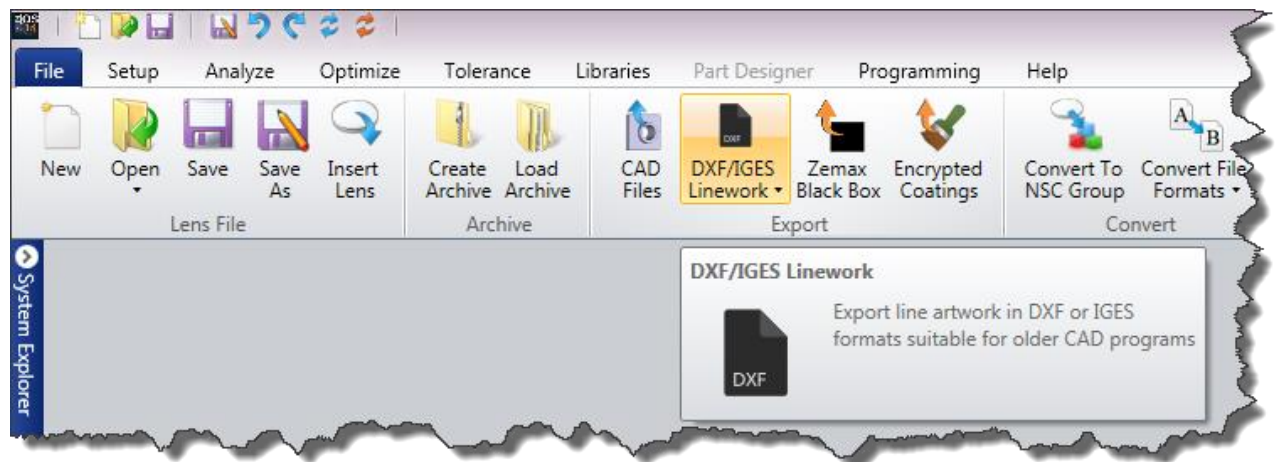
- Encircled Energy is now called Enclosed Energy as circles are not the only option ☺
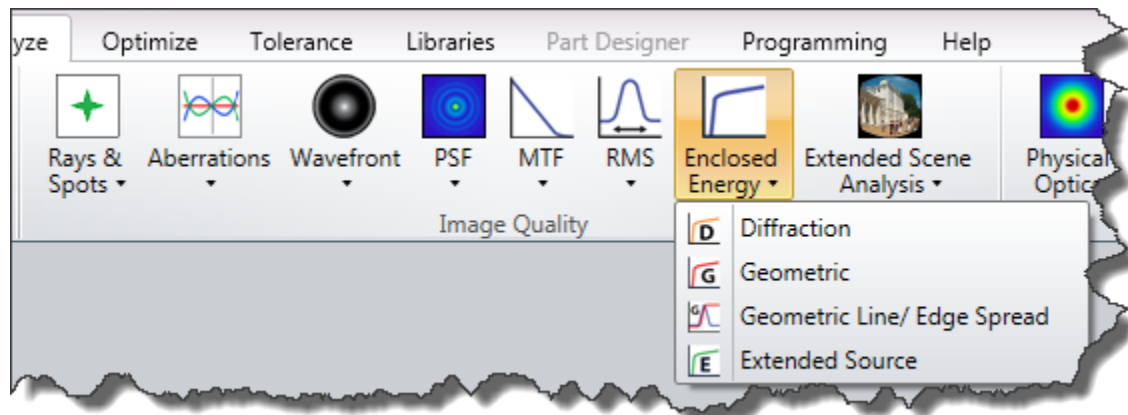
Zemax 13 Tools Menu
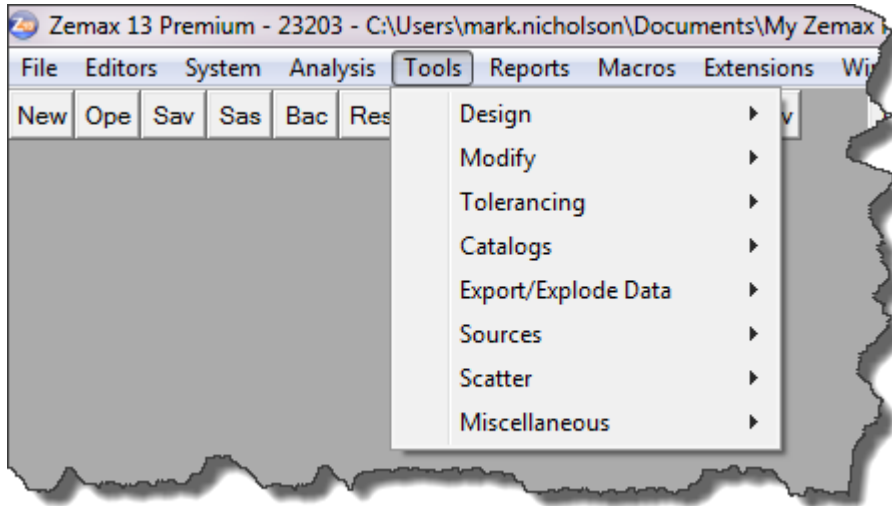
This entire menu is now obsolete, and items have either been moved to their own tabs or to the toolbars of the appropriate editor.

Tools…Design is now the Optimize tab
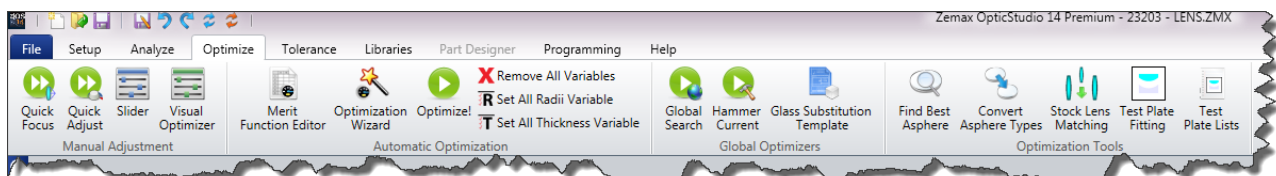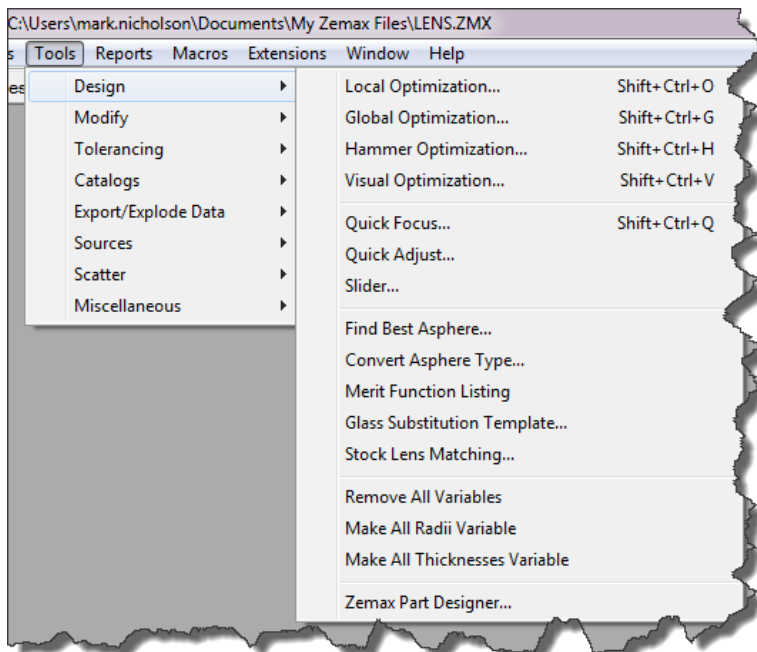
And the Zemax Part Designer is now a built-in capability with its own tab, which is available only in non-sequential mode, or if you have the Premium version of OpticStudio:
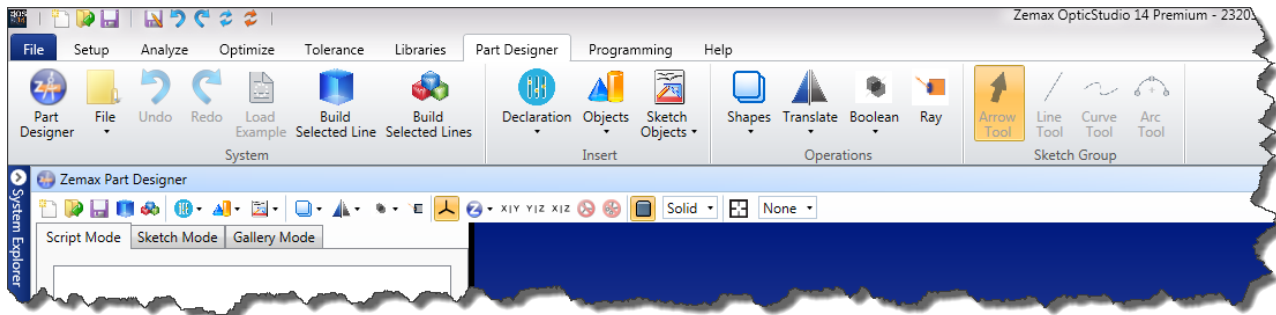
Tools…Modify has moved to the toolbar of the Lens Data Editor:
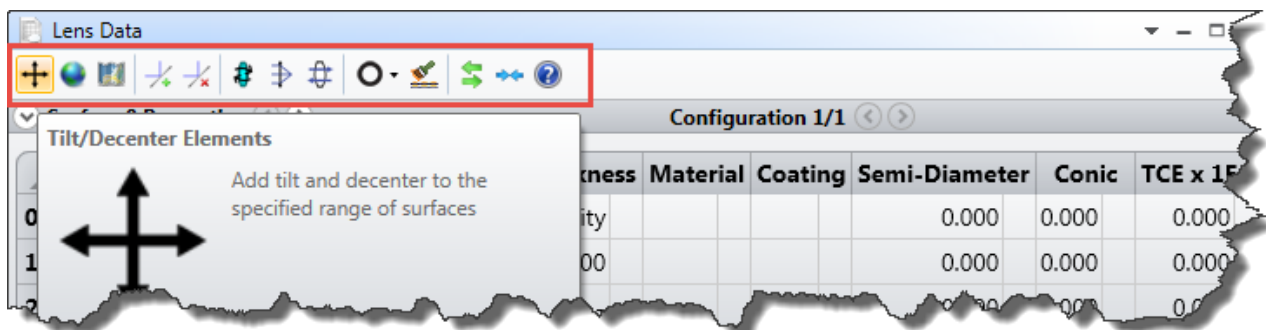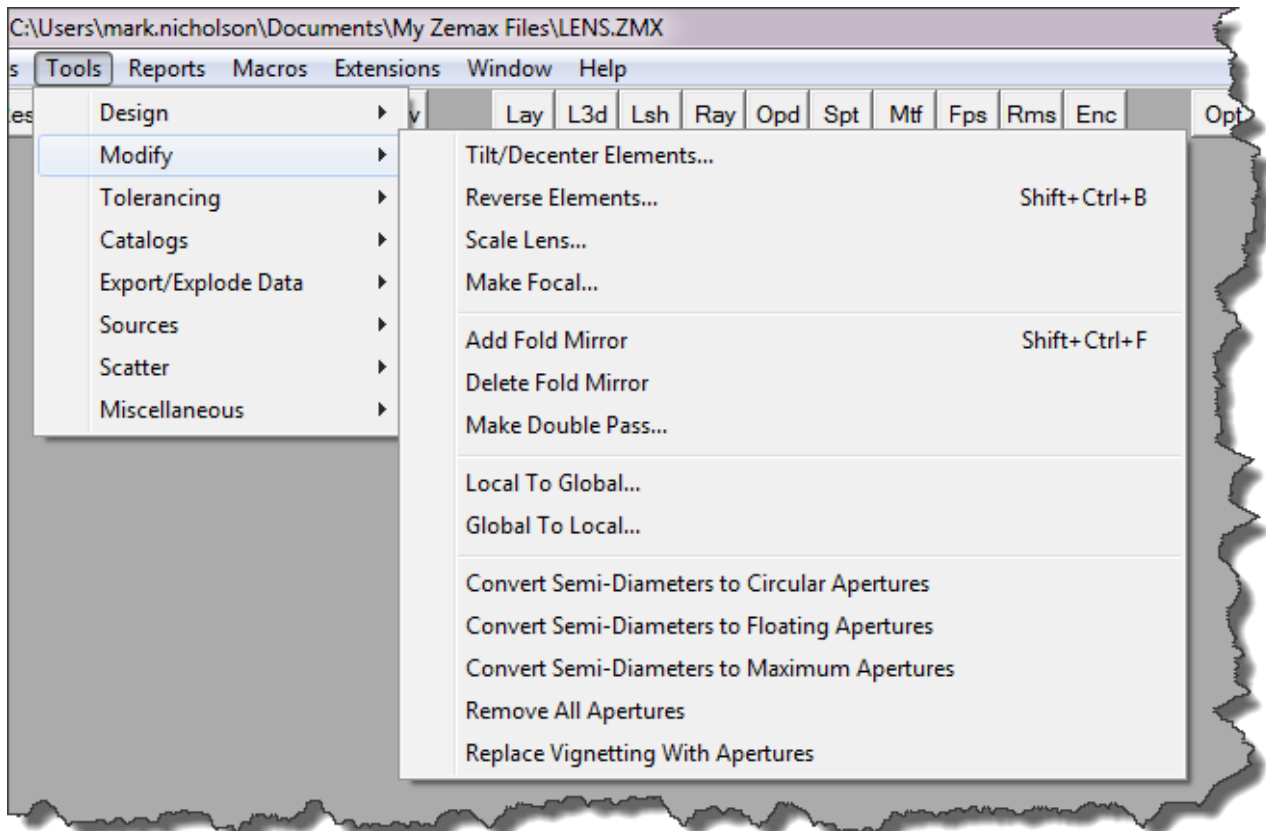
Tools…Tolerancing has become the Tolerance tab:
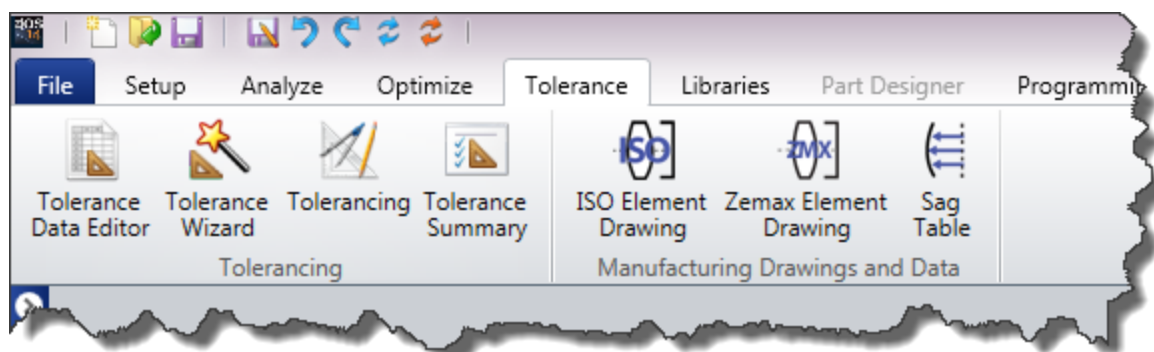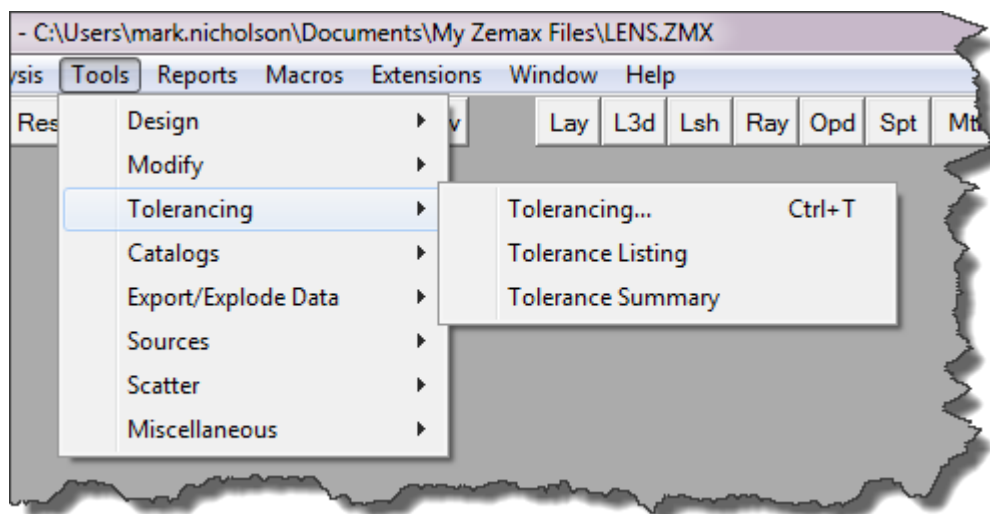
Tools…Catalogs has become the Libraries tab:
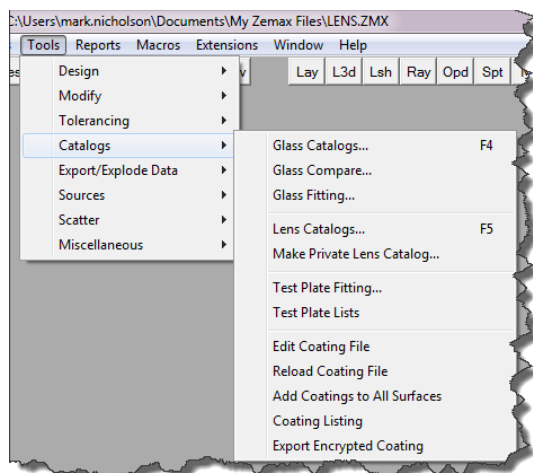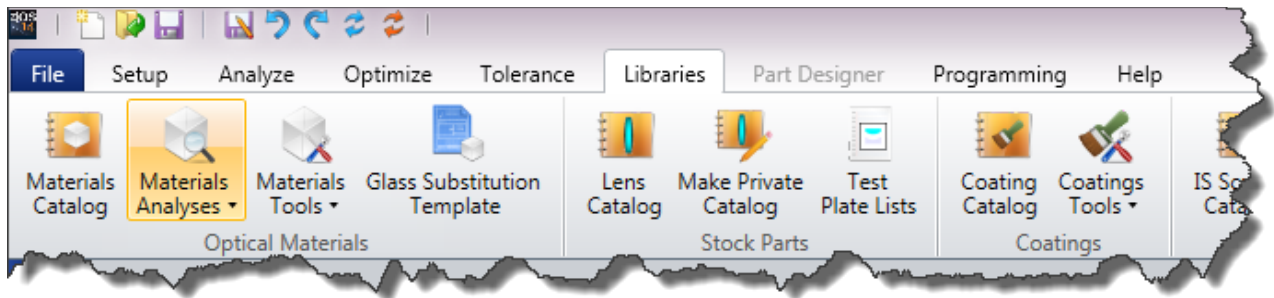
Tools…Export/Explode has moved to the Export and Explode groups on the File tab
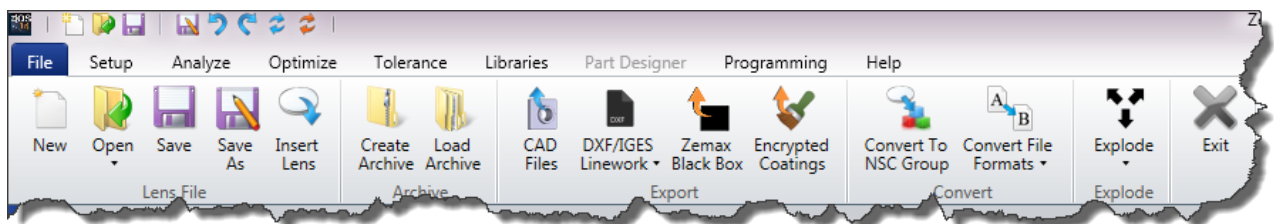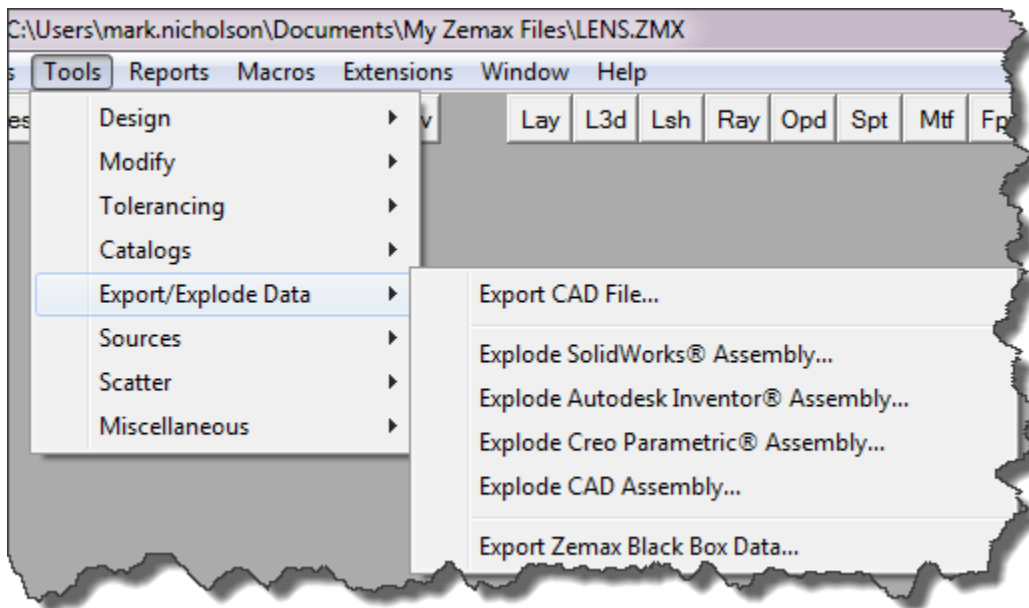
Tools…Sources has moved to the Sources group on the Libraries tab:

Radiant Source Models

The most accurate method of modeling the near-field and far-field output of real light sources in optical and illumination design software. Containing anular, spatial and optionally wavelength data, RSMs are detailed measurements of virtually any type of light source, including LEDs, fluorescent lights, arc, tungsten halogen and HID lamps
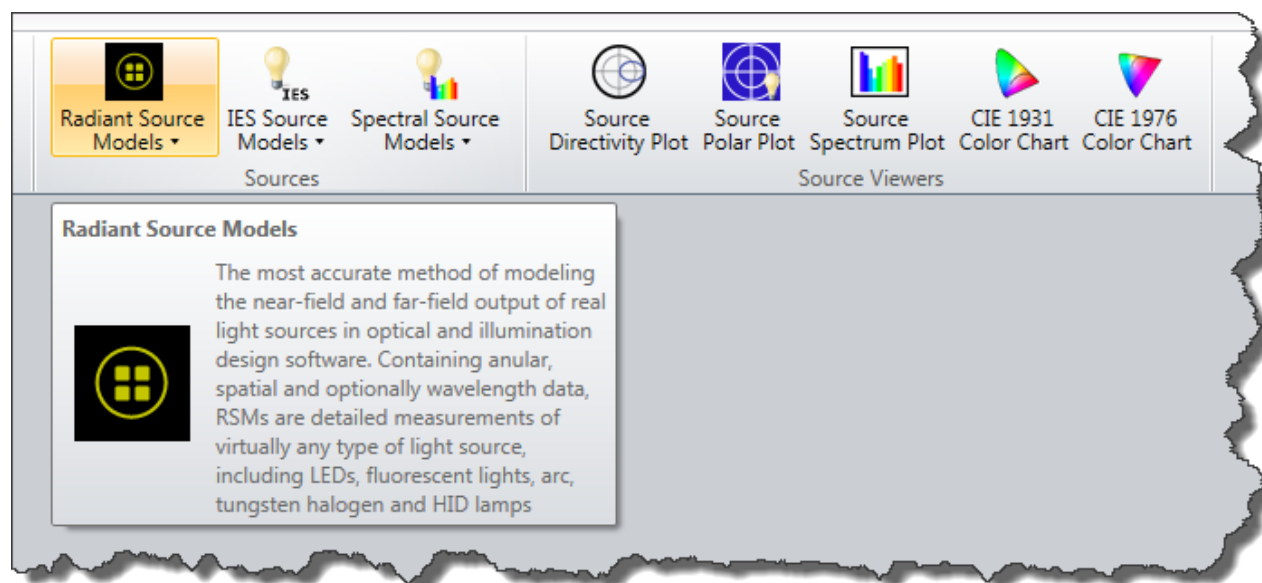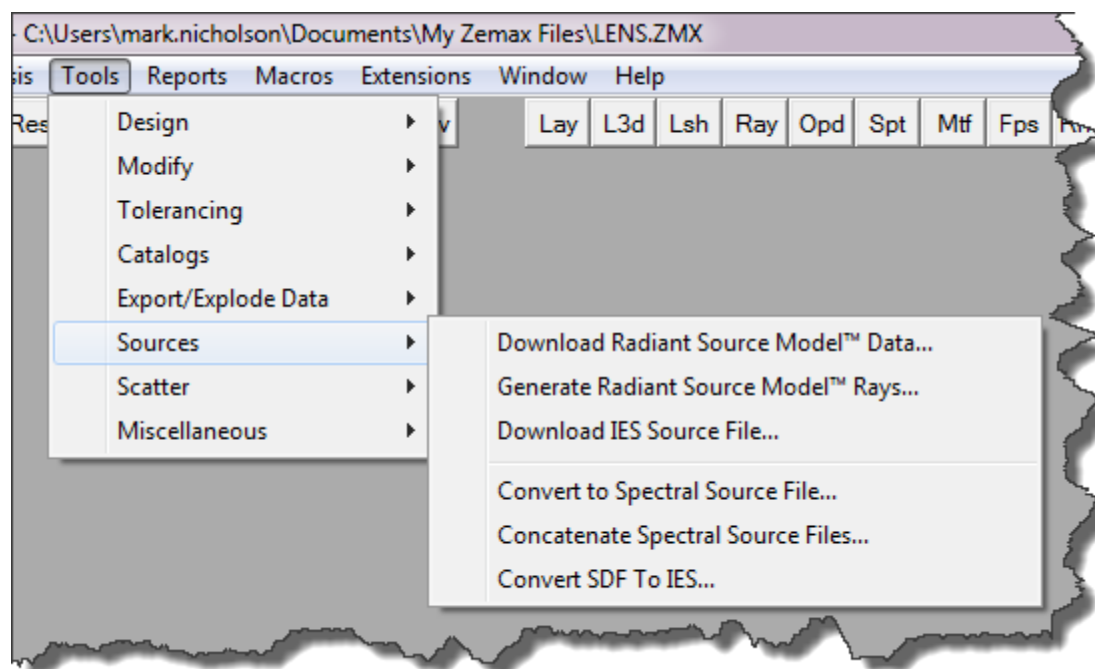
Tools…Scatter has moved to the Scatter group of the Libraries tab
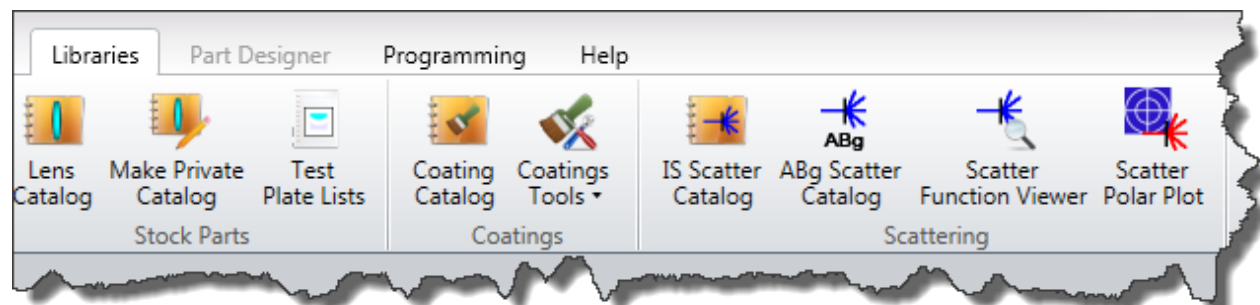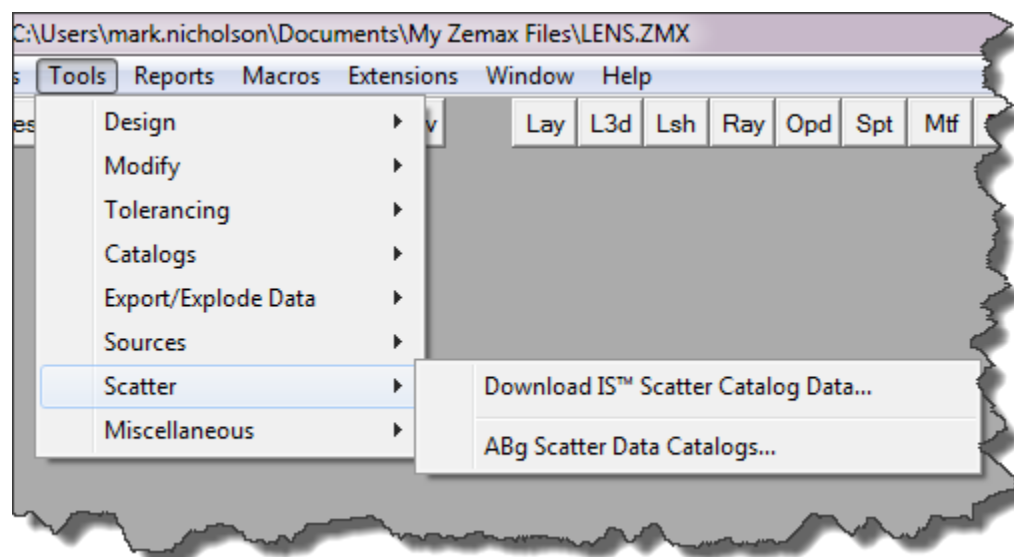
Tools…Miscellaneous is obsolete and its features have been placed in more meaningful locations:
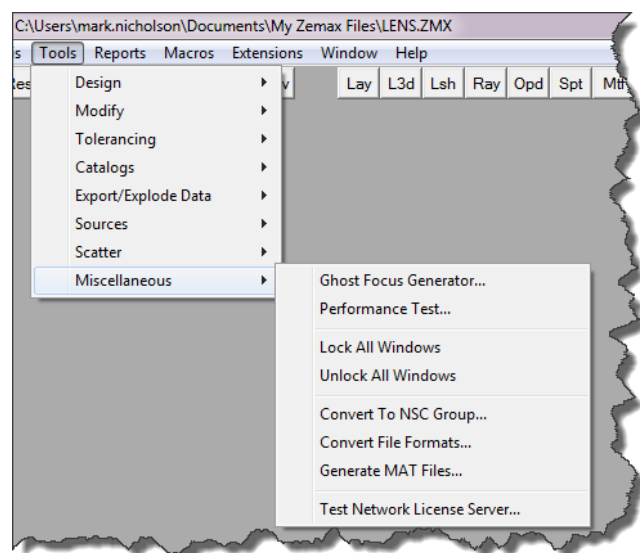
The Ghost Focus Generator is now on the Analyze tab, in the Applications Group
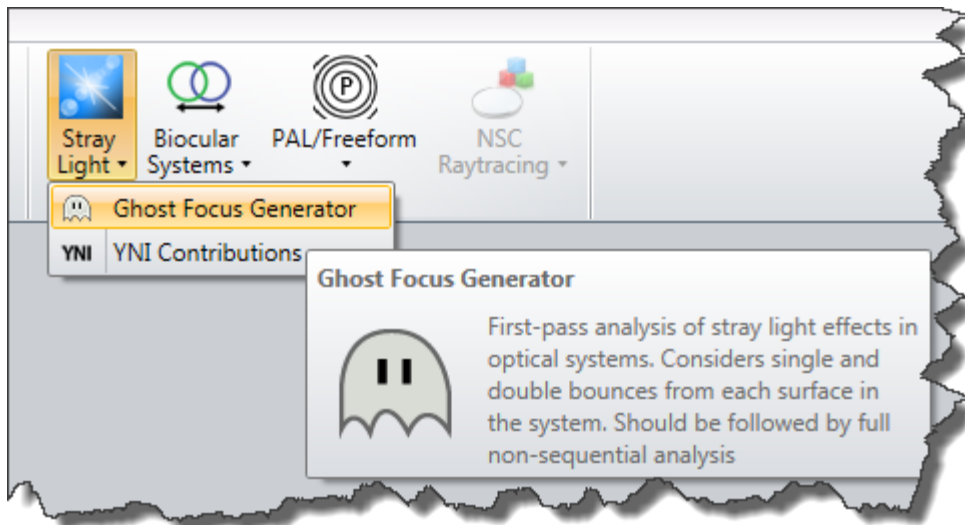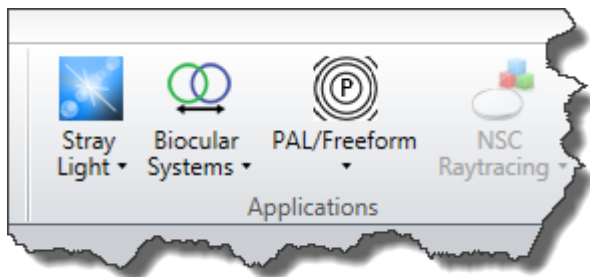
The Performance Test and Test Network Server are now in the Diagnostics group of the Setup tab:
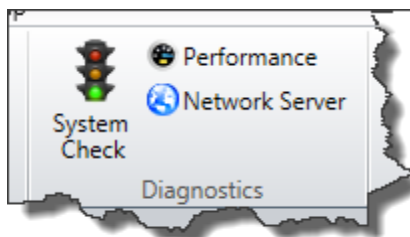
The file conversion and MatLab® export tools are now on the File tab Convert group:

Convert File Formats

Convert data files from other programs into the equivalent OpticStudio format

INT Zernike To OpticStudio DAT

INT Grid To OpticStudio DAT

INT Grid To OpticStudio GRD

Optiwave F3d To OpticStudio ZBF

MAT Convert To MAT

Convert To MAT

Reads the data from ZRD file or a ZBF file then writes it to native Matlab data file format.This feature requires the presences of a Matlab installation to run but does not require Matlab to run concurrently with OpticStudio

Zemax 13 Reports Menu

These are now in the Reports group on the Analyze tab:
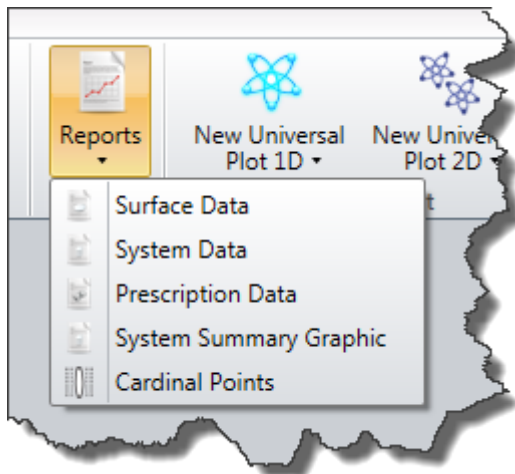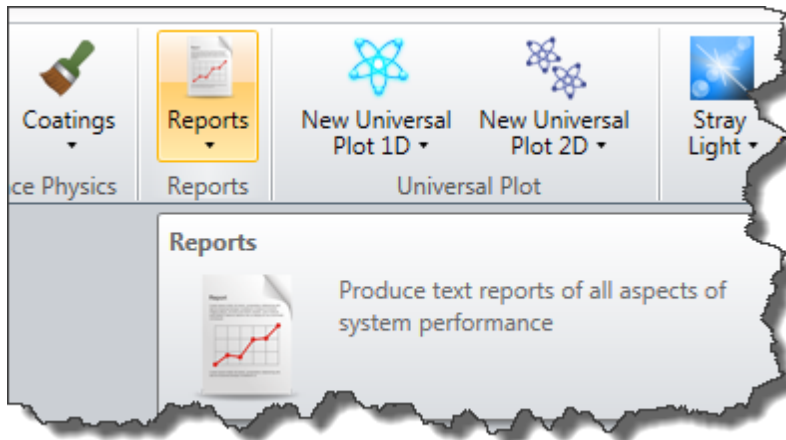
Note that the Report Graphic 4 and Report Graphic 6 have no equivalent in OpticStudio. These features were only supported by the Z13 graphics engine. A replacement feature will be incorporated in a future release.

Zemax 13 Macros Menu
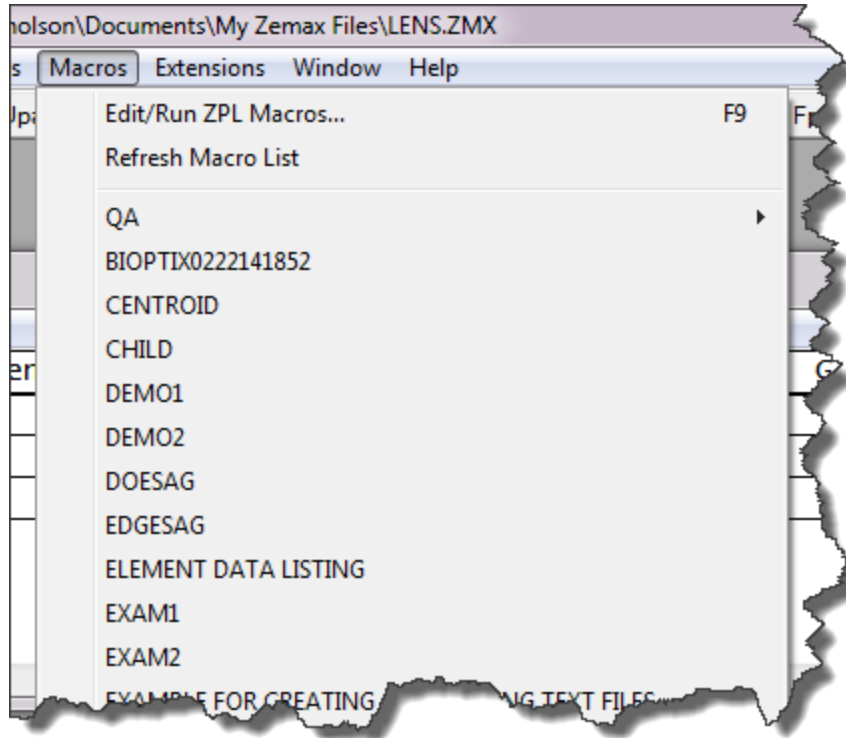
This is now in the ZPL Macros group of the Programming tab
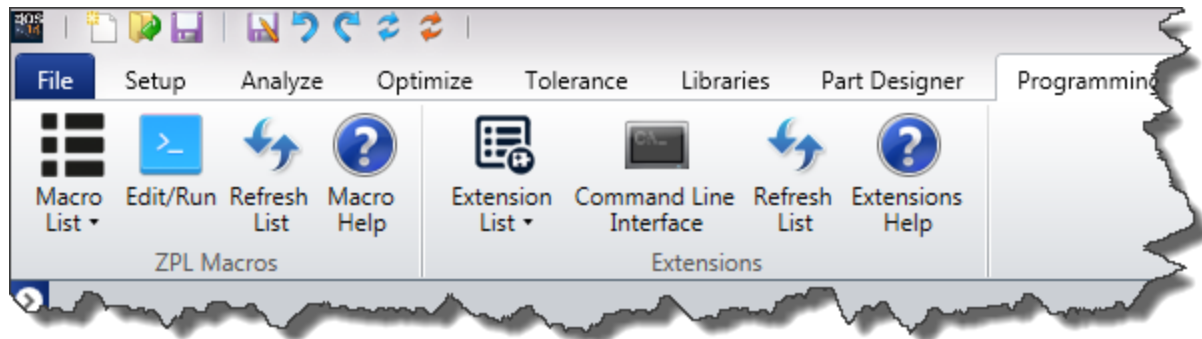
A built-in macro editor is now provided.

Zemax 13 Extensions Menu
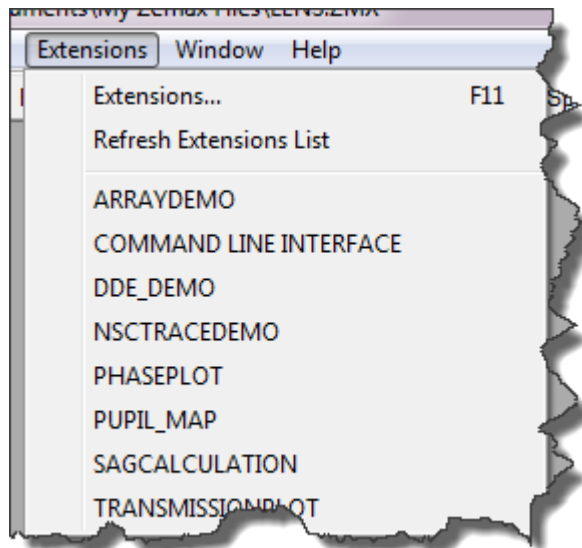
This is now in the Extensions group of the Programming tab
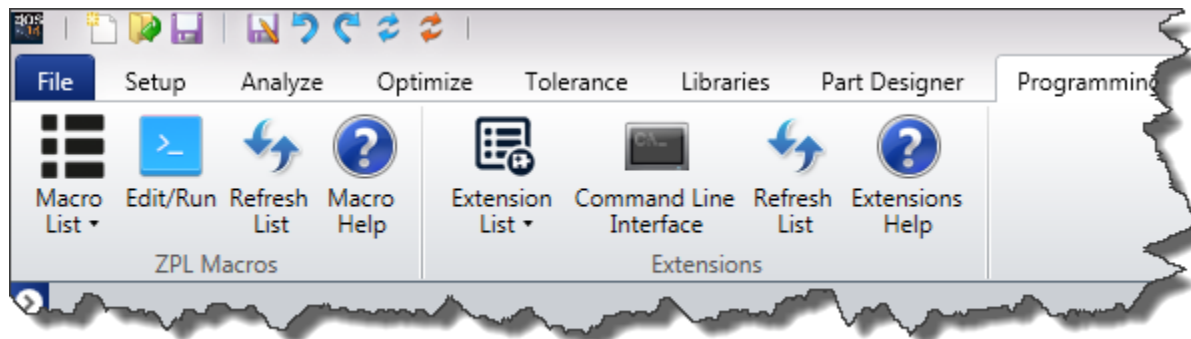
Zemax 13 Help Menu
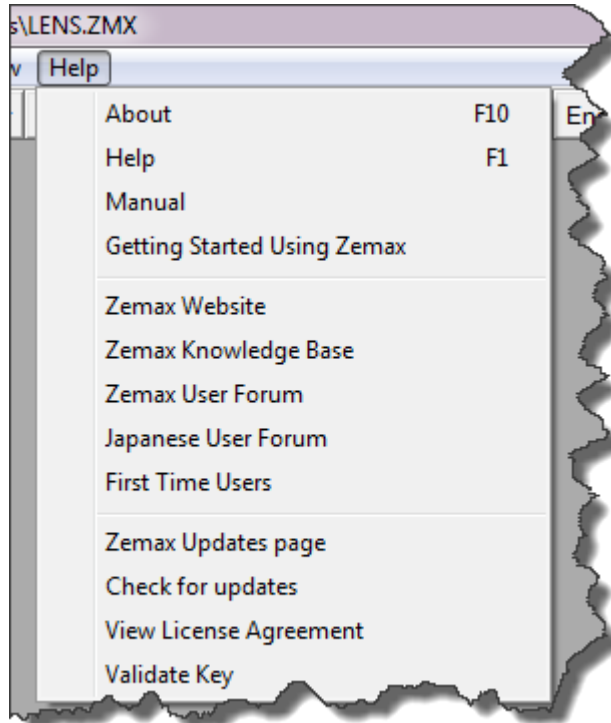
Now on the Help tab
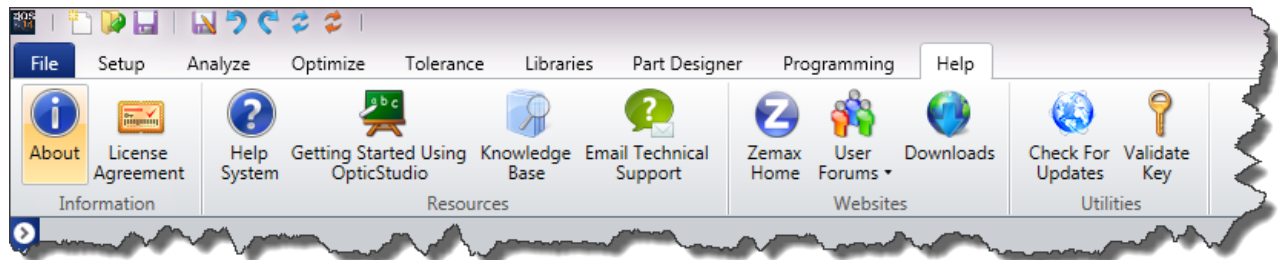
The Help system is now a searchable hyperlinked document, broken into three main groups:

1. The Getting Started Guide is intended as the place to start for anyone new to OpticStudio or the demo version of OpticStudio, to get a good grounding in the user interface.

2. The Technical Reference contains detailed descriptions of the various capabilities of the major components of OpticStudio

3. The In-Line Help contains all the feature-specific help which is available when you press the Help icon in any feature
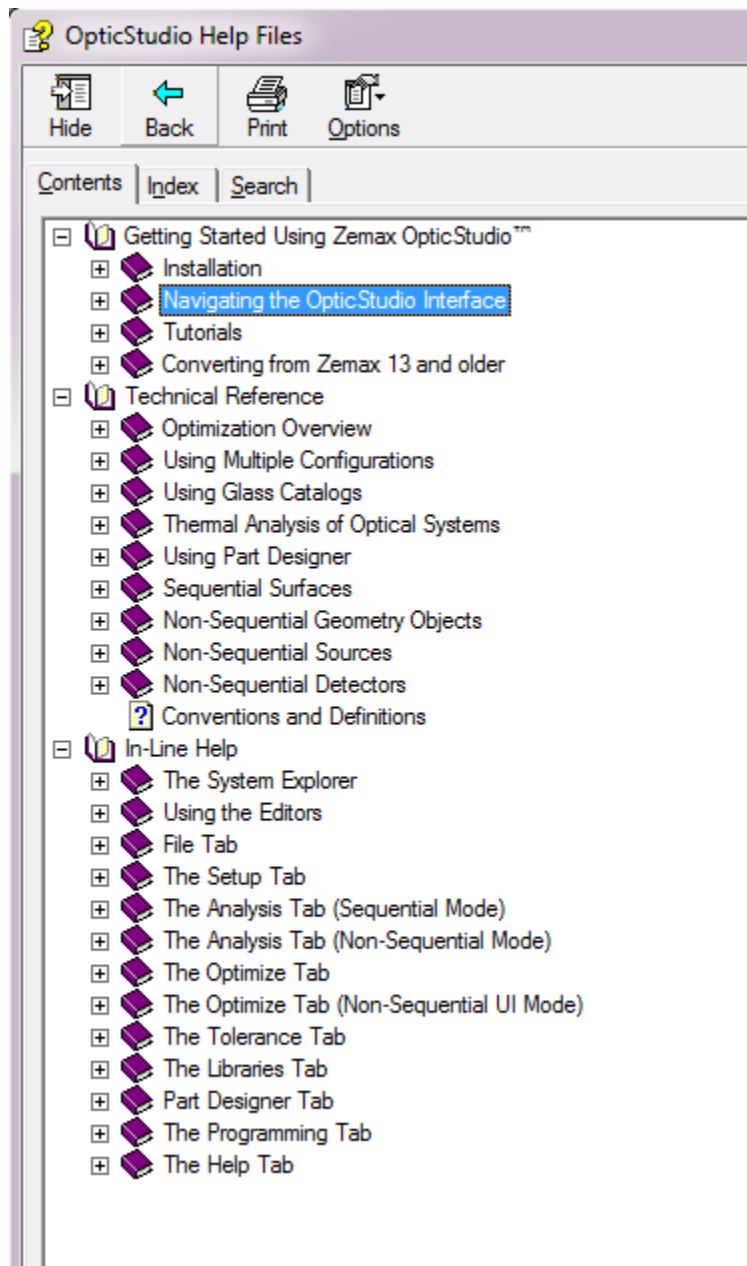
2D Analysis Windows

The Settings dialog may now be left open and changes may be auto-applied. Right-mouse-clicking now brings up a context menu. Data markers may be set on most windows, and the interactivity of all windows is greatly improved. Data is no longer recalculated when displayed differently: data is only recalculated if the calculation settings change.
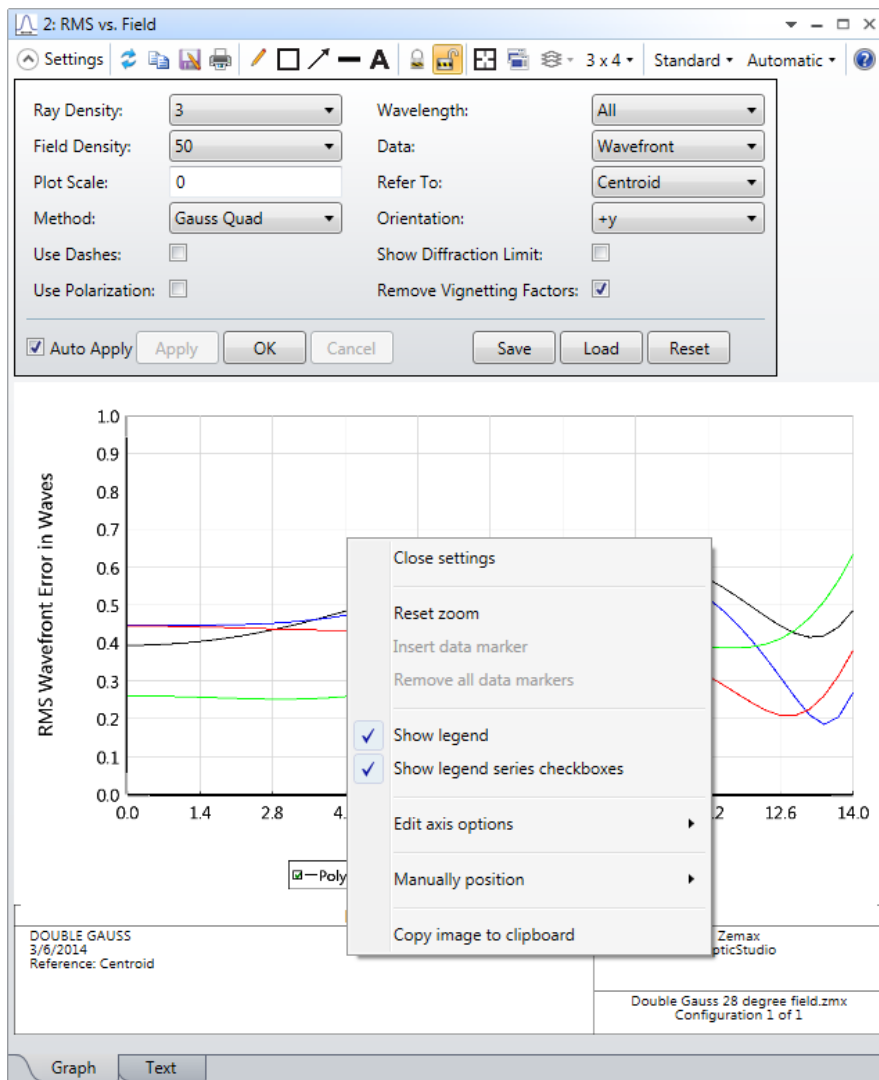
There is a wider range of options available via right-mouse-click, and a toolbar provides access to all tools associated with the window. The scale of the axes can be changed by selecting the sliders on each end of the axes. See the full discussion of Analysis windows in the Navigation section.

3D Analysis Windows

All the benefits of the 2D Analysis windows, plus the 3D projection can be rotated by holding down the left mouse button and dragging, zoomed by holding down the scroll wheel and panned by holding down the right mouse button. Again, see the full discussion of Analysis windows in the Navigation section.

Shaded Model

This has been completely re-written to provide a CAD-like user experience. See the full discussion in the <u>Navigation section</u>.

Editors

The editors have been completely re-written to make them more intuitive spreadsheets. Each editor how has a toolbar with all its associated tools on it. See <u>Using Spreadsheet Editors</u> for full details.

Using the Editors

This section gives a brief introduction to using the editors in Zemax.

These editors are:

- The Lens Data Editor
- The Non-Sequential Components Editor
- The Tolerance Data Editor
- The Multiple Configuration Editor
- The Merit Function Editor

Editor Windows Operations

The editor windows are used primarily for entering lens and merit function data. Each editor is similar to a spreadsheet, with rows and columns. The intersection of a row and a column forms a cell. If the editor is the active window (the one with a highlighted title bar) then one cell will be shown highlighted or in reverse colors. This cell is called the active cell, and it has what is called the input "focus". The inverted color of the cell is called the cursor, although it is not a cursor in the usual sense.

Having the input focus means that any data typed on the keyboard will be sent to the active cell. The exceptions are control commands such as cursor keys or control key combinations, which are sent directly to the main window. To modify the data in the active cell, type in the new data and press the Enter key when finished.

To add an incremental value to a cell, type a plus sign and then the increment, then Enter. For example, to change a 12 to 17, type "+5" and Enter. The "*" multiply and "/" divide symbols also work. To subtract a value, type a minus sign and a space followed by the value to subtract. The space is required to distinguish between subtraction and entering a negative number.

To modify a portion of the contents of a cell without retyping the entire value, either double-click the cell, or single click the cell and press the F2 function key. The left cursor, right cursor, home, and end keys can then be used to navigate within the cell for editing. The mouse may also be used to select and replace portions of text. Once changes have been made to the data in the cell, pressing Enter will complete editing and leave the cursor on that cell. Pressing the Up or

Down arrow keys will also complete editing, and move the cursor accordingly. Pressing Tab or Shift-Tab will complete editing and move the cursor right or left.

To copy (or paste) a cell value, either right-click and choose Copy Cell (or Paste Cell) from the right-click menu, or use the Ctrl + C (or Ctrl + V) keyboard shortcut. This copies (or pastes) the full precision of the cell value, even if only a few digits are displayed.

To select an entire row, either click on the row number, or use the keyboard shortcut and press Shift + Right arrow key when any cell in the row selected. To select multiple rows, either click and drag the mouse on the row number, or use the keyboard shortcut and press Shift + Up or Down arrow keys when a row is selected. Entire rows may be copied and pasted in the same manner as individual cells. Note that copying and pasting of rows is not supported between different releases of Zemax.

To abort editing of any cell, press the escape key.

The left, right, up and down cursor keys will move the cursor accordingly. Pressing the control key and the left, right, up and down keys simultaneously will move the editor display one page at a time in the appropriate direction. The Tab and Shift-Tab keys also move the cursor right and left.

The page up and page down keys move the cursor one page at a time. Control-page up and control-page down move the cursor to the top and bottom of the current column. The home and end keys will move the cursor to the first column first row and first column last row, respectively. Control-home and control-end will move the cursor to first and last row of the last column, respectively.

All editors support adjustable column widths. To change the size of the column, place the mouse pointer in the top row of the spreadsheet in the title area directly on one of the vertical column separator lines. When the mouse pointer changes to the resize icon, press the left mouse button and then drag the column width left or right to the desired size.

Undo, Redo, and Recover

There are three different states to the Undo capability in Zemax: None, Memory 1 Step, and Disk Multi Step. The Undo state is set in the Editors section of the Project Preferences dialog box described in "The Setup Tab".

Undo: None

If the Undo feature state is set to None, then no Undo capability is supported. Use this option on computers which do not have sufficient system memory or disk space to support the Undo feature.

Undo: Memory 1 Step

Zemax stores a copy of the current lens in memory before and after every edit or optimization. If Undo is selected, then the current lens is swapped with the previous lens. If Redo is then selected, the lenses are swapped again, which results in the edit being restored.

Memory 1 Step Undo is useful for restoring a lens when an accidental edit is made, or for restoring a lens to the prior state after an optimization. However, only a single Undo step is supported. The advantage of this option is speed; the saving of the prior lens in memory is so fast as to be unnoticeable.

Undo: Disk Multi Step

Zemax stores a copy of the current lens in a ZMX file on disk after every edit or optimization. These stored lens files are used to implement an infinite multi-step Undo feature, which allows the reversal of any change or series of changes made to the lens. Undo is useful for restoring a lens when an accidental edit is made, or for restoring a lens to the prior state after an optimization, or even several changes.

To reverse the changes made to a lens, select Undo button from the main Zemax toolbar. Any number of Undo's may be executed, back to the first edit made after loading the lens file. The Redo function reverses the last Undo.

Zemax maintains a folder of Undo files within the <data>\Undo folder (see "Folders"). The Undo files are automatically deleted whenever the file is saved, a new file is opened, or Zemax is normally terminated. If Zemax abnormally terminates, the operating system fails, the computer power is disrupted, or for any other reason the lens data is lost, Zemax may be able to recover the lost data by restoring the last Undo file. When Zemax starts, a check is made to see if any of the Undo files exist. Since these files are deleted during a normal termination, the presence of Undo files indicates a previous abnormal termination. Zemax will issue a warning message with the option of restoring the last Undo file. If restored, this file should immediately be saved in a new file name, since the old file name is not stored with the lens.

The Undo feature does slow down the operation of the Editors slightly, since every edit is followed by a save operation. The save does not slow down ray tracing or optimization speed, only lens data editing.

If more than one Zemax session is running simultaneously, each session will have its own undo files. However, to recover from a crash or abnormal program termination, the same number of Zemax sessions will need to be run to recover all the files. For example, if 2 sessions of Zemax are running, and the power fails, the first new session of Zemax will be able to recover the former first session file. A second session of Zemax will need to be launched to recover the former second session file.

Using the Lens Data Editor

The Lens Data Editor is the primary spreadsheet where the majority of the lens data will be entered. This data includes the radius of curvature, thickness, and glass for each surface in the system. Single lenses are defined by two surfaces (a front and a back), and the object and image also each require one surface. This fundamental data can be entered directly on the spreadsheet. When the Lens Data Editor is displayed, data can be entered on the spreadsheet by typing in the required values at the highlighted bar. Each column is labeled by the data type, and each row represents a single optical (or perhaps a dummy) surface. The cursor keys move the highlighted bar to whichever column is desired. Moving the cursor continuously to the right or left causes the display to scroll, which provides access to the other columns of data such as semi-diameters, conic constants, and "parameter" terms whose values depend upon the surface type. The display

will scroll left to right or right to left. The page up and page down keys will move the bar to the top and bottom of the display. The display will also scroll up and down if required, when the number of lens surfaces is sufficiently large.

See the Lens Data Editor.

Using the Merit Function Editor

The Merit Function Editor is used to define, modify, and review the system merit function for optimization.

See the Merit Function Editor and Optimization Overview.

Using the Multiple Configuration Editor

The Multi-Configuration editor is very similar to the Lens Data Editor and allows the user to set up different configurations of a lens file.

See the Multiple Configuration Editor and Using Multiple Configurations.

Using the Tolerance Data Editor

The Tolerance Data editor is used to define, modify, and review the system tolerance values.

See the Tolerance Data Editor and Tolerancing.

Using the Non-Sequential Components Editor

The NSC Editor is only used by the Non-Sequential Components surface type supported by Zemax-Professional and Zemax-Premium. The editor is very similar to the lens data spreadsheet editor. The editor can only be activated if a Non-Sequential Components surface is present in the Lens Data Editor, or if the program mode is set to Non-Sequential Mode. For information on changing the program mode, see "Program Mode".

See the Non-Sequential Components Editor for a detailed discussion of non-sequential components and ray tracing.

Technical Reference

This is the Technical Reference section of the Help Files. Click the "+" next to the "Technical Reference" header in the Contents pane to expand and navigate through this section.

Using Multiple Configurations

Zemax supports a very general capability for defining, analyzing, and optimizing optical systems which are used in multiple configurations. Multi-configurations are used to design zoom lenses, or to optimize lenses tested and used at different wavelengths, or optical systems used in various configurations, to name a few. Like other Zemax features, multi-configuration is well integrated. However, like tolerancing, it requires a little more care and practice to become proficient.

Zemax uses a substitution procedure for defining multiple configurations. The configurations are differentiated by different values for the same parameter. For example, in a zoom lens, the spacings between various elements may take on more than one value. Each set of values used together forms one configuration.

The first step

By far the most important step is to define one configuration using the normal Zemax mode first. It is a good idea to start with the most complex configuration first. If all of the configurations have the same number of elements, pick any one of them. Once you have the basic configuration defined, it is time to define new configurations which are variations of the first. The first configuration does not need to be optimized yet, you can optimize across configurations later.

Open the Multi-Configuration Editor from the Setup Tab. The spreadsheet which appears is the multi-configuration editor (MCE). Both configurations (columns) and operands (rows) can be inserted and deleted. The insert and delete keys will add or remove new operand rows. The data entered on the MCE will be saved automatically whenever you save your lens file.

See the Multiple Configuration Editor in the Editors section of the Setup Tab.

Multi-configuration operands table

To change the operand type, double click on the type column. A dialog box will appear where the type and number of the multi-configuration operand can be changed.

The operands are also summarized in the following table:

Type	Numbers 1,2,3	Description
AFOC	Ignored	Afocal Image Space mode. For more information, search the help files for "Afocal Image Space".
AICN	Surface, Object	iPartFactory Number for the Autodesk Inventor part. Search the help files for "Autodesk Inventor Part" for details.
APDF	Ignored	System apodization factor. See also APDT.
APDT	Ignored	System apodization type. Use 0 for none, 1 for Gaussian, 2 for cosine cubed. See also APDF.
APDX	Surface #	Surface aperture X- decenter. The surface must have a defined aperture (NOT semi-diameter).
APDY	Surface #	Surface aperture Y- decenter. The surface must have a defined aperture (NOT semi-diameter).
APER	Ignored	System aperture value. If the system aperture type is float by stop size this is the semi-diameter of the stop surface. See also SATP.
APMN	Surface #	Surface aperture minimum value. The surface must have a defined aperture (NOT semi-diameter). This same operand also works to control the first parameter of all surface aperture types, such as the X-Half Width on rectangular and elliptical apertures.
APMX	Surface #	Surface aperture maximum value. The surface must have a defined aperture (NOT semi-diameter). This same operand also works to control the second parameter of all surface aperture types, such as the Y-Half Width on rectangular and elliptical apertures.
APTP	Surface #	Surface aperture type. The integer values indicating the aperture type are 0-10 for none, circular aperture, circular obscuration, spider, rectangular aperture, rectangular obscuration, elliptical aperture, elliptical obscuration, user aperture, user obscuration, and floating aperture; respectively.
CADX	Surface #	Surface Tilt/Decenter after surface decenter x.
CADY	Surface #	Surface Tilt/Decenter after surface decenter y.
CATX	Surface #	Surface Tilt/Decenter after surface tilt x.
CATY	Surface #	Surface Tilt/Decenter after surface tilt y.
CATZ	Surface #	Surface Tilt/Decenter after surface tilt z.
CAOR	Surface #	Surface Tilt/Decenter after surface order. Use 0 for Decenter then Tilt, or 1 for Tilt then Decenter.

CBDX	Surface #	Surface Tilt/Decenter before surface decenter x.
CBDY	Surface #	Surface Tilt/Decenter before surface decenter y.
CBTX	Surface #	Surface Tilt/Decenter before surface tilt x.
CBTY	Surface #	Surface Tilt/Decenter before surface tilt y.
CBTZ	Surface #	Surface Tilt/Decenter before surface tilt z.
CBOR	Surface #	Surface Tilt/Decenter before surface order. Use 0 for Decenter then Tilt, or 1 for Tilt then Decenter.
CONN	Surface #	Conic constant.
COTN	Surface #	The name of the coating, if any, to be applied to the surface.
CPCN	Surface, Object	Family Table Instance Number for the Creo Parametric part. Search the help files for "Creo Parametric Part" for details.
CROR	Surface #	Coordinate Return Orientation. Use 0 for none, 1 for Orientation only, 2 for Orientation XY, and 3 for Orientation XYZ. For details, search the help files for "Using the Coordinate Return".
CRSR	Surface #	Coordinate Return Surface. For details, search the help files for "Using the Coordinate Return".
CRVT	Surface #	Curvature of surface.
CSP1	Surface #	Curvature solve parameter 1.
CSP2	Surface #	Curvature solve parameter 2.
CWGT	Ignored	The overall weight for the configuration. This number only has meaning relative to the weights in other configurations.
EDVA	Surface, Extra Data Number	The EDVA operand is used to assign multiple values to the extra data values. This operand requires 2 numerical arguments: the surface number and the extra data value number.
FLTP	Ignored	Field type. Use 0 for angle in degrees, 1 for object height, 2 for paraxial image height, 3 for real image height.
FLWT	Field #	Field weight.
FVAN	Field #	Vignetting factor VAN.
FVCX	Field #	Vignetting factor VCX.

FVCY	Field #	Vignetting factor VCY.
FVDX	Field #	Vignetting factor VDX.
FVDY	Field #	Vignetting factor VDY.
GCRS	Ignored	The global coordinate reference surface.
GLSS	Surface #	Glass.
GPEX, GPEY	obsolete	obsolete
GPJX	Ignored	Global Jones polarization vector component Jx.
GPJY	Ignored	Global Jones polarization vector component Jy.
GPIU	Ignored	Global polarization state "is unpolarized", 1 if polarization state is unpolarized, otherwise state is polarized.
GPPX	Ignored	Global polarization state phase x.
GPPY	Ignored	Global polarization state phase y.
GQPO	Ignored	Obscuration value used for Gaussian Quadrature pupil sampling in the default merit function. More details provided in "Selecting the pupil integration method".
HOLD	Ignored	Holds data in the multi-configuration buffer, but has no other effect. Useful for temporarily turning off one operand without losing the associated data.
IGNR	Surface #	Ignore This Surface status. Use 0 to consider the surface, and 1 to ignore the surface. For details, search the help files for "Ignore This Surface". If IGNR and IGNM operands are defined for the same surface, the one listed second will take precedence.
IGNM	First Surface , Last Surface	Sets Ignore This Surface status on a range of surfaces. Use 0 to consider the surfaces, and 1 to ignore the surfaces. For details, search the help files for "Ignore This Surface". If IGNR and IGNM operands are defined for the same surface, the one listed second will take precedence.
LTTL	Ignored	Lens title. The string length is limited to 32 characters.
MABB	Surface #	Model glass Abbe.
MCOM	Surface #	Surface comment.
MDPG	Surface #	Model glass dPgF.
MIND	Surface #	Model glass index.
MOFF	Ignored	An unused operand, may be used for entering comments.

MTFU	Ignored	MTF units. Use 0 for cycles/millimeter or 1 for cycles/milliradian.
NCOM	Surface, Object	Modifies the comment for non-sequential objects in the NSC Editor. The string value is limited to 32 characters.
NCOT	Surface, Object, Face #	Modifies the coating on each face for non-sequential objects in the NSC Editor.
NGLS	Surface, Object	The material type for non-sequential objects in the NSC Editor.
NPAR	Surface, Object, Parameter	Modifies the parameter columns for non-sequential objects in the NSC Editor.
NPOS	Surface, Object, Position	Modifies the x, y, z, tilt x, tilt y, and tilt z position values for non- sequential objects in the NSC Editor. The position flag is an integer between 1 and 6 for x, y, z, tilt x, tilt y, and tilt z, respectively.
NPRO	Surface, Object, Property	Modifies various properties of NSC objects. Property is an integer value indicating what data is controlled: 1 - Inside of object number 2 - Reference object number 3 - Do Not Draw Object (0 = no, 1 = yes) 4 - Rays Ignore Object (0 = never, 1 = always, 2 = on launch) 5 - Use Pixel Interpolation (0 = no, 1 = yes) 201-212 - User defined gradient index parameters 301-312 - User defined diffraction parameters for reflection 351-362 - User defined diffraction parameters for transmission 401-416 - User defined bulk scatter parameters 481, 482 - Bulk scatter mean free path and angle arguments. 500 - Media is birefringent. Use 0 for false and 1 for true, 501 - Birefringent mode. use 0-3 for ordinary and extraordinary rays, ordinary rays only, extraordinary rays only, and waveplate mode, respectively. 502 - Birefringent Reflections. Use 0 for refracted and reflected rays, 1 for refracted rays only, and 2 for reflected rays only. 503-505 - Birefringent crystal axis orientation x, y, and z.
PAR1	Surface #	Parameter 1. Obsolete, use PRAM instead.
PAR2	Surface #	Parameter 2. Obsolete, use PRAM instead.
PAR3	Surface #	Parameter 3. Obsolete, use PRAM instead.
PAR4	Surface #	Parameter 4. Obsolete, use PRAM instead.

PAR5	Surface #	Parameter 5. Obsolete, use PRAM instead.
PAR6	Surface #	Parameter 6. Obsolete, use PRAM instead.
PAR7	Surface #	Parameter 7. Obsolete, use PRAM instead.
PAR8	Surface #	Parameter 8. Obsolete, use PRAM instead.
PRAM	Surface, Parameter	Parameter value. This operand controls any of the parameters. For details, search the help files for "Parameter data".
PRES	Ignored	Air pressure in atmospheres. Zero means vacuum, 1 means normal air pressure. For details, search the help files for "Environment".
PRWV	Ignored	Primary wavelength number.
PSCX	Ignored	X Pupil Compress. Used for ray aiming; for details, search the help files for "Ray Aiming".
PSCY	Ignored	Y Pupil Compress. Used for ray aiming; for details, search the help files for "Ray Aiming".
PSHX	Ignored	X Pupil Shift. Used for ray aiming, for details, search the help files for "Ray Aiming".
PSHY	Ignored	Y Pupil Shift. Used for ray aiming, for details, search the help files for "Ray Aiming".
PSHZ	Ignored	Z Pupil Shift. Used for ray aiming, for details, search the help files for "Ray Aiming".
PSP1	Surface #	Parameter solve parameter 1 (the pickup surface). This operand requires 2 numerical arguments: the surface number and the parameter number.
PSP2	Surface #	Parameter solve parameter 2 (the scale factor).This operand requires 2 numerical arguments: the surface number and the parameter number.
PSP3	Surface #	Parameter solve parameter 3 (the offset).This operand requires 2 numerical arguments: the surface number and the parameter number.
PUCN	Ignored	Used for picking up a range of values from a previous configuration. If a positive integer configuration number is provided, then all values below the PUCN operand will be picked up from the configuration number specified. If the configuration value is negative, then a negative pickup while be used. If the configuration number is zero, then the values below the PUCN operand will not have pickup solves applied. Note two PUCN operands can be used to define the beginning and end of a range of values to be picked up. All specified configuration numbers must be less than the configuration the PUCN data is provided for.
PXAR	Surface #	Physical optics setting "Use X-axis Reference" (for details, search the help files for "Surface specific settings"). Use 0 for no, 1 for yes.

RAAM	Ignored	Ray aiming. Use 0 for off, 1 for paraxial, and 2 for real.
SATP	Ignored	System aperture type. Use 0 for Entrance Pupil Diameter, 1 for Image Space F/#, 2 for Object Space NA, 3 for Float By Stop Size, 4 for Paraxial Working F/#, 5 Object Cone Angle. See also APER.
SDIA	Surface #	Semi-diameter.
SDRW	Surface #	Modifies the do not draw this surface flag. Use 0 to draw and 1 to not draw.
STPS	Ignored	Stop surface number. The stop can be moved to any valid surface number (excluding the object and image surfaces) by specifying an integer argument for each configuration.
SWCN	Surface, Object	Configuration number for the SolidWorks part. More details provided in "SolidWorks Part tab".
TCEX	Surface #	Thermal coefficient of expansion.
TELE	Ignored	Telecentric in object space, 0 for no, 1 for yes.
TEMP	Ignored	Temperature in degrees Celsius. For details, search the help files for "Environment" .
THIC	Surface #	Thickness of surface.
TSP1	Surface #	Thickness solve parameter 1. For details, search the help files for "parameter solves".
TSP2	Surface #	Thickness solve parameter 2. For details, search the help files for "parameter solves"
TSP3	Surface #	Thickness solve parameter 3. For details, search the help files for "parameter solves"
UDAF	Surface #	User defined aperture file. Surface must use either a user defined aperture or user defined obscuration aperture type. For details, see the Aperture section of the Surface Properties.
WAVE	Wave #	Wavelength.
WLWT	Wave #	Wavelength weight.
XFIE	Field #	X-field value.
YFIE	Field #	Y-field value.

Comment about operands that define character strings

When using operands that define string values, such as LTLT, NCOM, and others, the string length is limited to 32 characters.

Defining the number of configurations

The number of configurations shown in the editor is changed by inserting or deleting configurations using the Insert or Delete toolbar buttons.

Defining each configuration

To define a multi-configuration operand, expand the list of operands by clicking the down arrow in the first column of the Multi-Configuration Editor. Click on the name of the operand you wish to select. Expand the operand properties by clicking the down arrow in the Operand Properties bar, and enter additional operand data if necessary.

For example, suppose you want to enter in a multiple value for the thickness of surface 5. Assume you want three different configurations. Insert 2 new configurations using the Insert button in the toolbar. Click on the left most column in row 1. Select "THIC" from the drop-down list of operand types. In the Operand Properties dialog, select "5" for the surface number. Now in each configuration column, enter the desired thickness in that configuration.

Ignoring surfaces

One application for the multi-configuration feature is to design systems that can be operated with or without certain elements being present. This can easily be handled by using the "Ignore This Surface" feature described in the "Type" section of the Surface Properties. The ignore surface setting can be controlled by the multi-configuration operand IGNR to ignore different surfaces in different configurations.

Changing configurations

To perform any analysis, proceed as usual. The program will use the current configuration for all calculations and graphics. See the "Active Configuration" definition in the Multiple Configuration Editor. To change configurations, double click on the column header at the top of the configuration you wish to change to on the MCE. The shortcut keys Ctrl-A and Shift-Ctrl-A also work.

Optimization with multi-configurations

Zemax will optimize multi-configuration data as readily as the conventional spreadsheet data. To make a multi-configuration parameter variable, place the cursor on the parameter, and press Ctrl-Z. This is a toggle; pressing Ctrl-Z again would eliminate the variability. When optimization is invoked, the optimization will automatically count the new variable. As many multi configuration variables as you like may be defined.

To select which (or all) configurations to optimize, open the Optimization Wizard from the Merit Function Editor. Depending on the user selection, Zemax will build the appropriate merit function for one or all of the available configurations. Optimization across configurations is performed using the CONF operand. This special operand changes the current configuration during the evaluation of the merit function. This means all operands defined after CONF will be relevant to the new configuration. CONF may be used multiple times in the operand sequence to evaluate various parameters.

Boundary constraints and other user entered optimization operands in multiple-configuration merit functions are evaluated, and therefore enforced, only in the configuration in which they are defined. For example, if a CONF 1 operand is followed by various operands such as EFFL or REAY, these will only be evaluated in configuration 1. To enforce these same operands in configuration 2, the same operands need to be repeated under the CONF 2 operand.

The advantage to this system is that the entered operands, or their respective targets or weights, may be different in each configuration. The disadvantage is the need to copy the operands that apply in more than one configuration to each configuration.

Suggestions for organizing multiple-configuration merit functions

There are two common different ways to organize a multiple-configuration merit function. The first way is to add the user defined operands within each CONF group of a default merit function:

```
CONF 1
User operands for configuration 1...
Default operands for configuration 1...
CONF 2:
User operands for configuration 2...
Default operands for configuration 2...
CONF 3:
...etc.
```

The other method is to build the default merit function, then add all the user operands at the top of the merit function to keep them all in one place:

```
CONF 1
User operands for configuration 1...
CONF 2
User operands for configuration 2...
CONF 3
User operands for configuration 3...
etc...
DMFS
CONF 1
```

```
Default operands for configuration 1...
CONF 2
Default operands for configuration 3...
CONF 3
Default operands for configuration 3...
etc...
```

Both merit functions will accomplish the same job, but the first method executes faster because the overhead in changing between configurations is reduced. The second method is easier to edit and maintain. Note the use of the DMFS operand after the user defined operands. This operand serves as a marker, so that when the default merit function is reconstructed, it will be appended after the DMFS operand, and the user entered operands are not lost.

Note that if you change the field angles, heights, or weights, or the wavelength values or weights on the multi-configuration screen you should rebuild the default merit function. When the default merit function is constructed, it uses the data for each configuration to determine the rays traced, and the appropriate weighting.

Solves for multi-configuration data

See the "Solve Types" section of the Multiple Configuration Editor in the Setup tab help file.

Using Glass Catalogs

The index of refraction for refractive surfaces and solids in Zemax may be defined several different ways. Most of this section is related to dispersion data as defined in the glass catalogs. For alternate means of defining index data, see "Alternate methods of defining dispersion data" later in this reference file. There are several standard catalogs supplied with Zemax, and custom catalogs may be created.

Zemax computes indices of refraction from formulas and coefficients entered into the glass catalogs. When you specify a glass name such as "BK7" in the glass column of the LDE, Zemax looks for the name in each of the currently loaded glass catalogs. If the glass is found, Zemax uses the coefficients for that glass, and then using the formula for the glass selected in the catalog, computes the indices at each of the defined wavelengths.

It is important to note that all Zemax glass catalogs assume that the index computed by the dispersion formulas is the relative index of refraction computed as a function of the relative wavelength. Relative means relative to air at 1.0 atmosphere and a reference temperature defined in the glass catalog. For details on how Zemax considers temperature, pressure, and the index of air see Index of refraction computation.

> **Zemax assumes that the index computed by the dispersion formulas is the relative index of refraction computed as a function of the relative wavelength.**

This method may seem more complex than directly entering in the indices of refraction, but the advantages are numerous. For one, the formulas are generally more accurate than user-entered data. Catalog data is more convenient as well, requiring the user to only supply the glass name. This is a particular advantage during the glass selection phase of the design. Additionally, any wavelength may be used, even if no explicit index data at that wavelength is available. The primary disadvantage is that the coefficients must be calculated, although this data is either readily available in catalogs or easily calculated. If you have the index data for some material not in the catalog, or if you feel your data is better than the catalog data, Zemax will compute the coefficients for you automatically; see "Fitting index data" later in this reference file.

Specifying which glass catalogs to use

This section describes the loading, editing, and managing of the glass catalogs. To specify the use of a particular catalog, use the Material Catalogs section of the System Explorer. By default, SCHOTT is the only catalog listed under "Catalogs To Use", but multiple catalogs may be used at one time. All other available catalogs are listed under "Catalogs Available", and are added and removed from the "Catalogs To Use" section via the arrow buttons.

All the listed catalogs must be in the Glass Folder folder (see Folders). The reason that glass catalog names are specified on this screen is that the catalog choice is stored with each lens separately. If you now save the lens, and reload it at a later time, the correct glass catalogs, and only those catalogs, will automatically be loaded.

Editing and reviewing glass catalogs

To edit or review data in an existing glass catalog, select Materials Catalog from the Libraries tab. Select the catalog name from those listed in the drop-down list on the dialog box. Once the catalog is selected, you may insert, cut, copy, paste, or modify data in the catalog, as is described in the following sections. You can save the newly modified catalog to either the same name or a new name. When editing the glass catalogs supplied with Zemax, be sure to save the modified data to a new name using the "Save Catalog As" button. This is important because future releases of Zemax may include an updated catalog which will be installed over the existing catalog, and any changes that had been made to the existing catalog will be lost.

To edit a large amount of data in the glass catalog, to create an extensive custom catalog, or to convert data in another format to the format used by Zemax, it may be easier to edit the glass catalog file directly. For information on the file format see "The AGF and BGF file formats" later in this reference file.

> **Be extremely careful when editing the glass catalogs; erroneous ray trace data will result if the catalog data is incorrectly modified.**

Description of catalog data

The glass catalog dialog box displays data about each individual glass. The data fields are described in the following table.

GLASS CATALOG DATA FIELDS

Item   **Description**

Catalog   Used to specify which of the .AGF format catalogs to display. The comment string to the right of the catalog name may be used to describe the catalog.

Glass   Used to specify which glass within the catalog to display data for.

Rename   In the event that a glass needs to be given a different name, the name of the currently selected glass can be modified in this field.

Formula   The dispersion of each glass is described by a formula. This control allows selection of which formula is used. If this setting is changed, then the dispersion data becomes invalid unless the appropriate coefficients are also entered. See "The glass dispersion formulas" later in this reference file.

Status   The status indicates the general availability of the glass. The available settings are Standard, Preferred, Obsolete, Special, and Melt. The status values are generally specified by the manufacturer. Standard glasses are generally available for purchase. Preferred glasses are usually frequently melted glasses, and more likely to be available upon demand. Obsolete glasses are no longer manufactured, but may be available. Special is a general category used to indicate a glass that does not fall into one of the other categories. Melt is a flag used by Zemax to indicate glasses that have been created in the catalog by the melt fit feature, see "Fitting melt data" later in this reference file.

Index Nd   The index at d-light, or 0.587 micrometers. Zemax does not use this number when computing the index of refraction. It is displayed solely for reference. The entry may be meaningless for some materials which do not transmit well in the region around 0.587 micrometers.

Abbe Vd   The Abbe value at d-light. This number is not used by Zemax when computing the index of refraction. It is displayed solely for reference.

Ignore Thermal Expansion   This switch allows accurate thermal modeling of non-solid materials, such as gasses and liquids, by allowing direct specification of the thermal coefficient of expansion in the lens data editor rather than in the glass catalog. Only the edge effects are considered; the radius of curvature and other thermal pickup solves will all use the adjacent material TCE, rather than the gas or liquid TCE.

Exclude Substitution   If checked, then this glass will not be selected during global optimization, conversion from model to real glasses, or be considered by the RGLA optimization operand.

Meta Material (Negative Index)   If checked, then the dispersion formula will be used to determine the absolute value of the index of refraction, but the sign will be reversed to yield a negative index of refraction. So-called Meta Materials have a negative index of refraction.

K1, L1... A0, A1..., A, B, C, etc.   The first eight rows in the center column of the dialog box display the dispersion coefficient data. The names of these coefficients changes depending upon the glass formula.

TCE   The TCE value is the thermal coefficient of expansion. This is a dimensionless parameter. The value displayed or entered needs to be multiplied by 1e-6 to yield the actual values.

Zemax currently uses the TCE value to model linear thermal expansion independent of the temperature range being used. The catalogs values supplied by the glass vendor are usually the TCE for the temperature range from -30 to +70 degrees Celsius, but Zemax will use the TCE value for any temperature range.

Temp   The reference temperature in degrees Celsius. This is the temperature of the air to which the relative wavelength and index data is referenced. This is usually the temperature of the glass at which the data was measured, but not always. See also Defining temperature and pressure.

D0, D1, D2, E0, E1, Ltk   These are the thermal coefficients used by the thermal analysis model. See [Index of refraction computation](#).

p   p is the material density in grams per cubic centimeter.

dPgF   The value is the deviation of the relative partial dispersion from the normal line.

Min, Max Wavelength   The minimum and maximum wavelengths in micrometers over which the dispersion formula returns valid index data.

Melt Frequency This integer value indicates the relative frequency with which the glass is melted by the manufacturer. The convention used is 1 means the glass is melted very frequently, 2 means less frequently, etc. up to a value of 5 to indicate a very infrequently melted glass.

Comment   This is an optional comment specific to the individual glass.

Relative Cost   This number is intended to indicate the approximate relative cost of the glass as compared to BK7. For example, a value of 3.5 would indicate the glass costs about 3.5 times as much as BK7 per pound. The number is only intended as a rough guide. The exact glass cost may vary with the quality, amount, form, and availability of the glass being purchased. Contact the glass vendor for more information.

CR, FR, SR, AR, PR   These are the glass codes that indicate how resistant the glass is to various environmental effects. The codes correspond to Climate Resistance (CR), Stain Resistance (FR), Acid Resistance (SR), Alkali Resistance (AR), and Phosphate Resistance (PR). Generally, the lower the number the more durable the glass is. If the value -1 or the symbol "?" is entered or displayed, no data is available. For a complete description of these codes and the test conditions used to measure them, contact the glass vendor.

Creating a new catalog

To create a new glass catalog, choose "Save Catalog As" and specify a new name. After the new catalog is created, any unwanted data may be removed using the "Cut Glass" button.

Note that Zemax does not allow any spaces in the names of glasses or glass catalogs.

Copying or moving glass catalog files

To copy or move a glass catalog file, copy or move only the file ending with the AGF extension. When Zemax reads the catalog with the AGF extension, a file with the same name will be created with a BGF extension. For example, Zemax

will read in the file SCHOTT.AGF and create a file named SCHOTT.BGF. The BGF version is a binary format version of the same data stored as text in the AGF file. Zemax will subsequently read the BGF version of the glass catalog because the binary version can be read much faster than the text version of the file. Zemax will automatically recreate the BGF version if the date stamp of the AGF file changes, or if the BGF file is deleted or missing, or if the Zemax version number changes.

The BGF file is of no value without the AGF file; so if a glass catalog must be copied, moved, or sent via email to another folder or location, copy the AGF file and not the BGF file. Zemax will automatically recreate the BGF whenever it is needed.

The glass dispersion formulas

The coefficients in the catalog are used in any one of several polynomial formulas that Zemax recognizes. There is also a dispersion formula described by the six-digit MIL number, but those indices are calculated directly from the MIL number entered on the spreadsheet. The MIL number formula is not part of the glass catalog, and so it will not appear. See "Using MIL number glasses" later in this reference file for a discussion of the MIL number glass formula. You may add new glasses to the currently loaded catalog if you have index data given in the form of one of the following equations. In all of the equations λ is in micrometers.

The Schott formula

The Schott constants of dispersion formula is

$$n^2 = a_0 + a_1\lambda^2 + a_2\lambda^{-2} + a_3\lambda^{-4} + a_4\lambda^{-6} + a_5\lambda^{-8}.$$

The required coefficients are available in most manufacturers' glass catalogs. Schott no longer uses this formula, but it is widely used by other glass manufacturers. See also "The Extended formula" later in this reference file.

The Sellmeier 1 formula

The Sellmeier 1 formula is

$$n^2 - 1 = \frac{K_1\lambda^2}{\lambda^2 - L_1} + \frac{K_2\lambda^2}{\lambda^2 - L_2} + \frac{K_3\lambda^2}{\lambda^2 - L_3}.$$

Coefficients for all three terms may be entered to describe the material, although fewer terms may be used. See also the Sellmeier 3 and Sellmeier 5 formulas.

The Sellmeier 2 formula

The Sellmeier 2 formula is

$$n^2 - 1 = A + \frac{B_1 \lambda^2}{\lambda^2 - \lambda_1^2} + \frac{B_2}{\lambda^2 - \lambda_2^2}.$$

Only two terms are used, there is no wavelength dependence in the numerator of the second term, and there is a constant term.

The Sellmeier 3 formula

The Sellmeier 3 formula is just like the Sellmeier 1 formula, with one additional term added:

$$n^2 - 1 = \frac{K_1 \lambda^2}{\lambda^2 - L_1} + \frac{K_2 \lambda^2}{\lambda^2 - L_2} + \frac{K_3 \lambda^2}{\lambda^2 - L_3} + \frac{K_4 \lambda^2}{\lambda^2 - L_4}.$$

The Sellmeier 4 formula

The Sellmeier 4 formula is:

$$n^2 = A + \frac{B \lambda^2}{\lambda^2 - C} + \frac{D \lambda^2}{\lambda^2 - E}.$$

The Sellmeier 5 formula

The Sellmeier 5 formula is just like the Sellmeier 3 formula, with one additional term added:

$$n^2 - 1 = \frac{K_1\lambda^2}{\lambda^2 - L_1} + \frac{K_2\lambda^2}{\lambda^2 - L_2} + \frac{K_3\lambda^2}{\lambda^2 - L_3} + \frac{K_4\lambda^2}{\lambda^2 - L_4} + \frac{K_5\lambda^2}{\lambda^2 - L_5} .$$

The Herzberger formula

The Herzberger expression is

$$n = A + BL + CL^2 + D\lambda^2 + E\lambda^4 + F\lambda^6 ,$$

$$L = \frac{1}{\lambda^2 - 0.028} .$$

The Herzberger formula is used mainly in the infrared spectrum.

The Conrady formula

The Conrady formula is

$$n = n_0 + \frac{A}{\lambda} + \frac{B}{\lambda^{3.5}} .$$

The Conrady formula is extremely useful for fitting to sparse data. For example, if you have only three index-wavelength pairs of data, fitting to the six-term Schott formula would yield meaningless data at the intermediate wavelengths.

The Handbook of Optics 1 formula

There are two similar formulas from the Handbook of Optics. The "Handbook 1" formula is:

$$n^2 = A + \frac{B}{(\lambda^2 - C)} - D\lambda^2 ,$$

The Handbook of Optics 2 formula

The "Handbook 2" formula is:

$$n^2 = A + \frac{B\lambda^2}{(\lambda^2 - C)} - D\lambda^2 .$$

The Extended formula

The Extended constants of dispersion formula is

$$n^2 = a_0 + a_1\lambda^2 + a_2\lambda^{-2} + a_3\lambda^{-4} + a_4\lambda^{-6} + a_5\lambda^{-8} + a_6\lambda^{-10} + a_7\lambda^{-12} .$$

This is similar to the Schott formula, with two additional terms added.

The Extended 2 formula

The Extended 2 constants of dispersion formula is

$$n^2 = a_0 + a_1\lambda^2 + a_2\lambda^{-2} + a_3\lambda^{-4} + a_4\lambda^{-6} + a_5\lambda^{-8} + a_6\lambda^4 + a_7\lambda^6 .$$

This is similar to the Extended formula, with a variation in powers of the wavelength in the last two terms.

The Extended 3 formula

The Extended 3 constants of dispersion formula is

$$n^2 = a_0 + a_1\lambda^2 + a_2\lambda^4 + a_3\lambda^{-2} + a_4\lambda^{-4} + a_5\lambda^{-6} + a_6\lambda^{-8} + a_7\lambda^{-10} + a_8\lambda^{-12} .$$

This is similar to the Extended formula, with another variation in powers of the wavelength.

General comments on using dispersion formulas

It is important to note that some publications use equations similar, but not identical to any of these expressions. It is often possible to rearrange the expressions to get them into the required form, and then recompute the required coefficients.

It is also a good idea to check the coefficients against a tabular listing of the index data available in many handbooks and publications. Use the dispersion plot or listing feature, or the prescription data report, which lists index data at each surface. If there are discrepancies, check the data you have entered carefully, and verify that the correct units and formula are being used.

Fitting index data

See also the discussion "Fitting melt data", below.

It is often the case that the materials you are designing with are in the catalog already. If they are not, you can enter the coefficients for the formulas described previously. As an alternative, Zemax will compute either the Schott, Herzberger, Conrady, or Sellmeier 1 dispersion formula coefficients for you. With the glass catalog dialog box displayed, click on "Fit Index Data" and the Fit Index Data dialog box appears. For a more comprehensive tool, see Glass Fitting.

On the left side of this screen is a two-column spreadsheet editor. Using the mouse, enter in the wavelength (in micrometers) and index data you have. The more data you enter, the more accurate will be the fit.

If you have more data than will fit in the spreadsheet, use the data most closely representing the wavelength region of interest. At least three points are required to get a good fit if you are using the Conrady formula, six or more (and preferably twelve to fifteen) for the Schott, Herzberger, or Sellmeier 1 formulas. Select the formula to use by selecting the formula name from the drop-down list. You may want to try each of the models in turn, to see which gives the lowest residual.

The RMS error is the RMS fit error between the given data, and the index data generated using the resulting fit coefficients. The max error is the largest error between the fit and any one data point. Both numbers can be compared to the magnitude of the index of refraction; which is of course dimensionless. Because the Sellmeier 1 formula has non-linear coefficients, the fitting is iterative, and this formula takes much more computer time to fit the data than the other formulas do.

Now move the cursor over to the "Name" field, and enter the material name for the catalog. Select "Fit" and Zemax will compute the optimal coefficients. The residual RMS error, and the maximum single-point error, are listed on the bottom of the display. To enter this data in the currently loaded catalog, select "Add to catalog". Zemax will issue a message verifying that the glass was saved.

> **When a glass is added to the catalog, the transmission data, if any, needs to be added as described in the next section. Otherwise, the default internal transmission of 1.0 is used at all wavelengths.**

The index and wavelength data may also be saved to a text file for later use, and loaded for fitting again by selecting the appropriate buttons. The text file may also be edited outside of Zemax, and then loaded for fitting.

Fitting melt data

See also the discussion "Fitting index data" above.

It is important to understand that the index of refraction values computed by Zemax, or listed in the catalog of the glass manufacturer, are average values for the index of refraction over a large number of "melts" or batches of the glass. A specific piece of glass from one melt will deviate from the catalog or nominal values slightly. The deviation is typically small, but the difference between the nominal and actual index values may be important for some systems.

Usually, when quality optical glass is shipped from the manufacturer, a data sheet will accompany the glass which indicates the index of refraction for the supplied glass at a few wavelengths, either as an offset from the nominal catalog values or as the measured index directly. Typically 3-5 wavelength-index data points are provided. This data is called "melt" data because it is specific to the batch of glass melted at one time.

The Melt Data tool available from the glass catalog dialog box is a handy utility for converting the limited melt data provided into a usable new glass type in the glass catalog.

There are a maximum of 8 wavelength-index points allowed for melt data. If you have more than 8 points, use the "Fit Index" tool described in the previous section. The minimum number of points allowed is 3, however, at least 4 and preferably 5 points should be used to get a good melt fit. The wavelength range defined by the melt index data should be as broad as possible, and should at a minimum cover the intended range of wavelengths to be used for ray tracing through the melt glass. In all cases, the fitted data should be inspected carefully for accuracy before it is used.

The Melt Data tool supports the following controls:

Glass: The name of the nominal glass in the selected glass catalog.

Melt Name: The name of the new glass to be created. The default is the nominal glass name with "_MELT" appended on. The name length may not exceed 20 characters.

Fit only these wavelengths: If checked, the melt index fit will only be done over the wavelength range defined by the minimum and maximum wavelengths of the provided melt data points. This allows much more precise fitting of the glass data, however, the melt glass cannot be used outside of this wavelength range. If unchecked, Zemax will attempt to extrapolate the data (see "Discussion of melt fitting method" later in this reference file) to create a melt fit that is valid over the entire wavelength spectrum of the original glass data. This is the key difference between fitting index data (see "Fitting index data" earlier in this reference file) and fitting melt data.

Formula: The dispersion formula to use for the new melt glass. Choose either Schott, Herzberger, Conrady, or Sellmeier 1. The default value is the Schott formula unless the nominal glass uses one of these formulas; in which case the same formula as the nominal glass will be used.

Use: This box turns "on" and "off" each row of data.

Wavelength: The wavelength in micrometers for the index values to be entered.

Nominal: The index of refraction at the defined wavelengths using the nominal glass dispersion.

Actual: The actual measured index from the melt data. Note if the actual value is edited, the "delta" value is automatically adjusted to keep the data consistent.

Delta: The difference between the actual and nominal index of refraction. Note if the delta value is edited, the "actual" value is automatically adjusted to keep the data consistent.

Fit/Insert: Choosing this button starts the fitting process as described below. Cancel: Aborts the melt fitting process.

After the fitting is finished, the new melt glass will be inserted in the catalog, the catalog will be saved, and a report summarizing the fit will be presented.

Discussion of melt fitting method

The problem with fitting melt data is the generally low number of points available; typically 3-5. Most fitting routines need at least 8 points for good accuracy. So, the problem is to extrapolate from a few points the variation in index over a large enough number of points to fit the resulting dispersion accurately. The accuracy of the resulting melt fit at the defined data points depends largely upon the extent of the wavelength range. Greater accuracy is achieved if "Fit only these wavelengths" (see discussion above) is checked, at the expense of a reduced spectral range over which the fit is valid.

Zemax does melt fitting using the following algorithm:

First, a fit of the actual dispersion data is computed using the Conrady formula. The Conrady formula is used because it is stable and reasonable when as few as three points are defined.

Then a Conrady fit of the nominal data is computed using only the defined wavelength points.

A large number of index points covering either the entire usable wavelength range (if "Fit only these wavelengths" is unchecked) or the wavelength range defined by the melt points (if "Fit only these wavelengths" is checked) of the nominal glass is generated. To each nominal index value, an offset is added which is the difference between the two Conrady fits that were generated using only the melt data wavelengths.

Finally, the resulting data is fit using the selected formula (not necessarily the Conrady). This is the final fit inserted into the catalog.

After the melt fitting is finished, a report summarizing the melt fitting process is presented. Check this report carefully before using the new melt glass!

Zemax automatically copies over all transmission, density, cost factor, and other data from the nominal glass to the melt glass.

Check the generated melt fitting report for index accuracy carefully before using the new melt glass!

Defining Transmission Data

Selecting the "Transmission" button invokes the transmission data editor within the glass catalog. Transmission refers to the intensity transmittance of light that depends upon the thickness of the glass as well as the wavelength. Zemax models the transmitted intensity using Beer's law:

$$t = e^{-\alpha\tau},$$

where α is the absorption coefficient and τ is the path length through the glass. The parameter α generally depends upon wavelength and has units of inverse length. See Polarization Analysis for information on polarization ray tracing and transmission.

The transmission is defined in the glass catalog by a series of 3 numbers: the wavelength in micrometers, the intensity transmittance, and the reference thickness in mm. For example, the transmission of a glass at 0.35 micrometers may be 0.65 for a thickness of 25 mm. Multiple data points may be defined in the transmission data editor. Internally, Zemax converts the data to "per mm", and interpolates between defined wavelengths. If ray tracing is being performed at a wavelength outside of the defined wavelengths, then the data for the closest wavelength is used; otherwise, Zemax performs a linear interpolation.

To save the transmission data to a text file, use the "Save To File" button. The file format is three columns of data: wavelength in micrometers, intensity transmittance, and reference thickness in mm. Data in this same format may be read from a file and placed in the catalog using the "Load From File" button. To save the modified data directly in the original glass catalog, use the "Save Catalog" button.

Not all of the glasses listed in the supplied catalogs have valid transmission data, especially for infrared materials and other non-commercial glass types. If transmission data is supplied by the manufacturer, it is usually included. If no reliable data is available, or if the data has been omitted, the default internal transmission of 1.0 is used at all wavelengths.

The Transmission Data editor ignores all data after the first entry with a wavelength (Lambda) of zero. Only the data entered before the first zero wavelength is considered.

Modeling gases and liquids

Once a material is defined in the glass catalog, Zemax uses the TCE specified for that material to determine the thermal expansion of the radius, center thickness, and other data for the surfaces using the material. However, if the material is not a solid, but is instead a gas or a liquid, then generally the thermal expansion is not governed by the material properties, but is instead determined by the edge thickness of the mounting material.

In this special case, Zemax needs to use the TCE supplied in the Lens Data Editor to define the mounting material properties, rather than the TCE supplied in the glass catalog. This can be accomplished by setting the "Ignore Thermal Expansion" switch for the material in the glass catalog.

Finding a glass quickly

The fastest way to view the data for any material or glass is to click once on the name of the material in the Lens Data Editor, then select Materials Catalog from the Libraries tab. The correct glass catalog and material will be displayed.

Glass catalog sources

The data for the glass catalogs included with Zemax are generally provided by the manufacturers. The data has been converted to Zemax format. The following vendors have supplied data for the Zemax glass catalogs:

GLASS VENDORS

Catalog	Vendor	Website
ANGSTROMLINK	Fiber Optic Center, Inc.	www.focenter.com
ARCHER	Archer OpTx	www.archeroptx.com
ARTON	JSR Corporation	www.jsr.co.jp
CDGM	Chengdu Guangming	www.cdgmgd.com
CORNING	Corning France Optical Division	www.corning.com
HEARAEUS	Heraeus Quarzglass GmbH & Co. KG	www.heraeus.de
HIKARI	Hikari Glass USA, Inc.	www.hikariglass.com
HOYA	Hoya Optics, Inc.	www.hoya.co.jp

IRPHOTONICS	IRphotonics Inc.	www.irphotonics.com
ISUZU	Isuzu Glass Co. Ltd.	www.isuzuglass.com
LIGHTPATH	LightPath Technologies	www.lightpath.com
LZOS	Russian optical glass fabricator	www.lzos.ru
NHG	Hubei New Huaguang Information Materials Co., Ltd	www.hbnhg.com
OHARA	Ohara Corporation	www.oharacorp.com
PILKINGTON	Pilkington Special Glass Limited	www.pilkington.com
RAD_HARD	Schott Glass Technologies	www.schott.com
RPO	Rochester Precision Optics	www.rpoptics.com
SCHOTT, SCHOTT_IG	Schott Glass Technologies	www.schott.com
SUMITA	Sumita	www.sumita-opt.co.jp
TOPAS	Polyplastics Co., Ltd.	www.polyplastics.com
UMICORE	Umicore Electro-Optic Materials	www.optics.umicore.com
ZEON	Zeon Chemicals L.P.	www.zeonchemicals.com

Although the data contained in the glass catalogs is generally reliable, it is always possible for errors to be made during translation or editing of the data. *It is absolutely crucial that all index data be verified for accuracy by the end user! This is especially true where fabrication of the optics is being considered.*

The catalog data may be in error, and it needs to be verified before it can be trusted.

The data contained in other catalogs provided with Zemax, such as the BIREFRINGENT, INFRARED, and MISC catalogs have been compiled from published sources as described in the following table. Some materials, such as calcite, are defined in more than one catalog. Materials may be included in the catalog which are not referenced in the table, or the source is specified as "unknown". These materials should be especially scrutinized before trusting the data accuracy.

BIREFRINGENT CATALOG DATA SOURCES

Material	**Source**
ADP	Handbook of Optics Vol. II
AG3ASS3	Handbook of Optics Vol. II
AGGAS2	Handbook of Optics Vol. II
AGGASE2	Handbook of Optics Vol. II
AL2O3	Handbook of Optics Vol. II
ALN	Handbook of Optics Vol. II
BATIO3	Handbook of Optics Vol. II
BBO	Handbook of Optics Vol. II
BEO	Handbook of Optics Vol. II

CALCITE	Handbook of Optics Vol. II
CAMOO4	Handbook of Optics Vol. II
CAWO4	Handbook of Optics Vol. II
CDGEAS2	unknown
CDS	Handbook of Optics Vol. II
CDSE	Handbook of Optics Vol. II
CUGAS2	Handbook of Optics Vol. II
KDP	Handbook of Optics Vol. II
LAF3	Handbook of Optics Vol. II
LINBO3	Handbook of Optics Vol. II
LIO3	Handbook of Optics Vol. II
LIYF4	unknown
MGF2	M.J. Dodge, "Refractive Properties of Magnesium Flouride," Applied Optics, Vol. 23, No. 11: p1980 (1985)
PBMOO4	Handbook of Optics Vol. II
QUARTZ	Handbook of Optics Vol. II
SRMOO4	Handbook of Optics Vol. II
TAS	Handbook of Optics Vol. II
TE	Handbook of Optics Vol. II
TEO2	Handbook of Optics Vol. II
YV04	Handbook of Optics Vol. I
ZNGEP2	Handbook of Optics Vol. II
ZNO	Handbook of Optics Vol. II

INFRARED CATALOG DATA SOURCES

Material	Source
ALN	Handbook of Optics Vol. II
ALON	Handbook of Optics Vol. II
AMTIR1	Amtir Spec Sheet
AMTIR3	Amtir Spec Sheet
BAF2	Handbook of Optics Vol. II
BEO	Handbook of Optics Vol. II
CAF2	Handbook of Optics Vol. II
CALCITE	Handbook of Optics Vol. II
CDSE	Handbook of Optics Vol. II

CDTE	Handbook of Optics Vol. II
CLEARTRAN	Spec Sheet (Rohm & Haas)
CLEARTRAN_OLD	Spec Sheet (Morton)
CSBR	Handbook of Optics Vol. II
F_SILICA	The Infrared & Electro-Optical Systems Handbook V. III
GAAS	Amtir Spec Sheet
GEO2	Handbook of Optics Vol. II
GE_LONG	JOSA, 47, 244 (57): 48, 579 (58)
GE_OLD	JOSA, 47, 244 (57): 48, 579 (58)
GERMANIUM	Handbook of Optical Constants of Solids
IRG2	Schott Spec Sheet
IRG3	Schott Spec Sheet
IRGN6	Schott Spec Sheet
IRG7	Schott Spec Sheet
IRG9	Schott Spec Sheet
IRG11	Schott Spec Sheet
IRG15	Schott Spec Sheet
IRG100	Schott Spec Sheet
KBR	ISP Optics data sheet (www.ispoptics.com)
KCL	ISP Optics data sheet (www.ispoptics.com)
KRS5	Handbook of Optics Vol. II
LIF	Handbook of Optics Vol. II
MGF2 11: p1980 (1985)	M.J. Dodge, "Refractive Properties of Magnesium Fluoride," Applied Optics, Vol. 23, No.
MGO	Handbook of Optics Vol. II
NACL	The Infrared & Electro-Optical Systems Handbook V. III
PBF2	Handbook of Optics Vol. II
SAPPHIRE	The Infrared & Electro-Optical Systems Handbook V. III
SILICON	Handbook of Optics Vol. II
SRF2	Handbook of Optics Vol. II
SRTIO3	Handbook of Optics Vol. II
ZBLA	Handbook of Optics Vol. II
ZBLAN	Handbook of Optics Vol. II
ZNGEP2	Handbook of Optics Vol. II
ZNSE	Handbook of Optics Vol. II
ZNS	The Infrared & Electro-Optical Systems Handbook, V.III

MISC CATALOG DATA SOURCES

Material	Source
ACRYLIC	Handbook of Optics Vol. II
BASF5	Laikin, Lens Design
BASF55	Laikin, Lens Design
CAF2	Handbook of Optics Vol. II
CDS	Handbook of Optics Vol. II
COC	Hoechst Celanese Spec sheet
CR39	The Photonics Design and Applications Handbook, Optical Plastics: Properties and Tolerances, H. D. Wolpert, pp. H-300 - H-307, (1991).
KDP	Handbook of Optics Vol. II
LAF3	Handbook of Optics Vol. II
LIYF3	Handbook of Optics Vol. II
PMMA	Handbook of Optics Vol. II
POLYCARB	Handbook of Optics Vol. II
POLYSTYR	Handbook of Optics Vol. II
PYREX	Laikin, Lens Design
QUARTZ	Handbook of Optics Vol. II
SAN	Handbook of Optics Vol. II
SEAWATER	Laikin, Lens Design
SILICA	Handbook of Optics Vol. II
TEO2	Handbook of Optics Vol. II
TYPEA	Laikin, Lens Design
VACUUM	Laikin, Lens Design
WATER	Handbook of Optics Vol. I

Obsolete catalog data

Changes in environmental protection laws have required the discontinuation of the manufacture of many optical glasses between the years 1990 - 2000. Roughly 2 out of 3 glasses which were listed in earlier catalogs of the various glass manufacturers have been discontinued. In some cases, glasses were reformulated to comply with new environmental restrictions. These new glasses may have identical or similar names as the old glasses, depending upon the glass vendor. The reformulated glasses may have different index data than the glasses they replace.

Obsolete glasses still exist in optical shops and may be used for new designs, if a supply can be located. Also, many older designs using the old glasses need to be modeled and ray traced, especially when designing new optics to work in

harmony with the existing optical system. For these reasons, data for obsolete glasses are still provided with the catalogs distributed with Zemax. Obsolete glasses are indicated by the "obsolete" status displayed in the glass catalog dialog box.

Because some new glasses may have the same name as old glasses, although the exact composition may have changed, optical engineers need to be especially vigilant about checking the index data predicted by the software against the melt sheets of the glass which will actually be used.

Never blindly trust the accuracy of the index computed using catalog coefficients. There are numerous sources of potential error, such as the measurement of the original sample, the fitting of the data, the typing in of the data in the manufacturers catalog, and then retyping into Zemax, and finally, the Zemax code itself.

The AGF and BGF file formats

Zemax stores glass catalog data in two file formats, called the ANSI Glass Format (AGF) and the Binary Glass Format (BGF). The glass catalog data supplied with Zemax is in the AGF format, and AGF files may be used when glass catalog data needs to be modified or created by the glass manufacturer. When Zemax runs, AGF files are automatically converted, if required, to BGF files. The BGF files are only used by Zemax to speed up the loading of the glass catalogs, and should never be created, edited, or distributed. Zemax will create or update a BGF whenever required. The AGF file is the "master" file used to define glass catalog data

The AGF file consists of a header line followed by a series of records, one for each glass. The header line format is:

```
 CC <Catalog Comment>
```

The mnemonic CC stands for Catalog Comment, and any comment string up to 140 characters may be placed here to define a comment for the catalog as a whole. Each subsequent record consists of a two letter mnemonic followed by one or more data items separated by spaces. For a description of these data see "Description of catalog data" also in this reference file. The format of a single record is:

```
 NM <glass name> <dispersion formula #> <MIL#> <N(d)> <V(d)> <Exclude Sub> <status>
<melt freq>
 GC <Individual Glass Comment>
 ED <TCE (-30 to 70)> <TCE (100 to 300)> <density> <dPgF> <Ignore Thermal Exp>
 CD <dispersion coefficients 1 - 10>
 TD <D0> <D1> <D2> <E0> <E1> <Ltk> <Temp>
 OD <rel cost> <CR> <FR> <SR> <AR> <PR>
 LD <min lambda> <max lambda>
 IT <lambda> <transmission> <thickness>
 IT <lambda> <transmission> <thickness>
 ... multiple IT lines may follow
```

Comments on the individual fields follow.

NM: Glass name and other data. The dispersion formula number is 1 for Schott, 2 for Sellmeier 1, 3 for Herzberger, 4 for Sellmeier 2, 5 for Conrady, 6 for Sellmeier 3, 7 for Handbook of Optics 1, 8 for Handbook of Optics 2, 9 for Sellmeier 4, 10 for Extended, 11 for Sellmeier 5, and 12 for Extended 2. The MIL# is provided for back compatibility and is not used, but a placeholder value must be provided. The nd and vd values are also provided for reference but are not used. The "exclude sub" flag is 0 for no and 1 for yes. Status is 0 for Standard, 1 for Preferred, 2 for Obsolete, 3 for Special, and 4 for Melt. Melt Freq is an integer between 1 and 5 to indicate the relative frequency of melting by the manufacturer.

GC: Glass comment. Any text string up to 140 characters long may be placed here.

ED: Extra Data. The values are the TCE in the -30 to 70 degree Celsius range, the TCE in the 100 to 300 degree Celsius range (this value is currently not used), the density, dPgF, and the Ignore Thermal Expansion flag, which is 0 for no and 1 for yes.

CD: Coefficient Data. Up to 10 coefficients may be provided. The meaning of these coefficients depends upon the dispersion formula used.

TD: Thermal Data. These are the D0, D1, D2, E0, E1, Ltk, and reference temperature values.

OD: Other Data. The relative cost, CR, FR, SR, AR, and PR values. For these values, -1 should be specified if the data is not available.

LD: Lambda Data. The minimum and maximum wavelength in micrometers over which the dispersion formula is valid.

IT: Internal Transmittance. These lines are the internal transmittance. Each line defines a wavelength, transmission, and thickness in millimeters. The wavelengths must be in ascending order. Up to 100 points may be defined for each glass record.

Alternate methods of defining dispersion data

There are alternate methods to define dispersion data that do not involve the glass catalogs and dispersion formulas. These methods are described in the following sections.

Using MIL number glasses

MIL number glasses are those described by a six-digit number, such as 517640 for BK7. The first three digits in the MIL number is the d-light index minus one, without the decimal place. The last three digits is the Abbe V- number times 10. To use a MIL number glass, type the 6 digit number in directly for the glass name.

Zemax uses a formula for computing the index at each defined wavelength based upon the index and Abbe number defined by the MIL number. The formula is based upon a least-squares fit of coefficient data of many typical glasses.

Typically, the index data calculated is accurate to roughly 0.001. At wavelengths in the deep UV or infrared the index value becomes less reliable. MIL number glasses are generally inferior substitutes for the constants of dispersion (or other) models for the glass, however they are useful if no other data is available.

> **Note MIL number glasses are an approximation, although usually a very good approximation in the visible range. Outside the visible wavelength range, such as in the ultraviolet or infrared, the MIL number glass is not accurate and should not be used.**

It is important to note that the indices calculated from the six-digit MIL number are not the same as those calculated from the glass catalog, even if the MIL number you are using corresponds to a glass in the catalog. Index data is calculated directly from the MIL number entered on the main screen; not from the glass catalog data, even if a glass with that MIL number is in the catalog.

Because any glass with a six digit name is assumed to be a MIL number glass defined by Zemax's internal equations, six digit numbers may not be used for glass names defined in the glass catalogs. MIL number glasses are not used by the glass substitution feature (see Using glass substitution).

Using table glasses

Table glasses are defined by data stored in a text file ending in the extension ZTG (for Zemax Table Glass). The file must be placed in the Glass Folder folder (see Folders to edit this path). The file format is:

```
! Comment line
DENSITY grams_per_cc
wavelength_1  index_1  transmission_1  thickness_1
wavelength_2  index_2  transmission_2  thickness_2
wavelength_3  index_3  transmission_3  thickness_3
etc..
```

Any blank line or a line starting with the "!" symbol is assumed to be a comment and is ignored.

The DENSITY data line defines the density of the glass in grams per cubic centimeter. If the DENSITY data line is omitted the density is assumed to be zero.

The wavelength must always be in units of micrometers and should be listed in ascending order. The index of refraction must be a positive value. The transmission is the internal transmittance of the glass and should be a positive value. The thickness is the thickness of glass in millimeters for the specified internal transmittance. The transmission and thickness values are used to determine the internal transmission per unit length. If both the transmission and thickness values are omitted the material is assumed to be 100% transmitting.

Up to 1200 lines of wavelength data may be defined in the file. At least 5 data points should be defined to provide a reasonable spline fit. If fewer than 5 points are provided, dummy points are added before and after the specified wavelength range to provide a smooth fit. The data should be checked for accuracy and reasonable fitting by using the dispersion plot data, see Dispersion Diagram.

To use the table data, enter the file name in the glass column for the desired surface, including the extension. For example, if the file name is "MYGLASS.ZTG" then "MYGLASS.ZTG" should be entered in for the glass name. Note the total name, including the extension, may not exceed 20 characters.

When computing the data for a specific wavelength, Zemax will use a cubic spline interpolation if the wavelength is between the shortest and longest wavelengths defined in the ZTG file. For wavelengths outside of the defined range, the data for the nearest data point is used; no extrapolation of the data is performed.

The advantage of a table glass is the index at the defined points is exactly the index specified in the table, unlike fitted functions which may deviate from the data being fitted at the data points.

Table glasses are not used by the glass substitution feature (see Using glass substitution). Table glasses are always assumed to be defined at a pressure of 1.0 ATM. No thermal properties are defined for table glasses. If thermal analysis is important, a catalog glass should be used instead. For information on converting measured index and thermal data into a catalog glass see Glass Fitting.

Using model glasses

Zemax can idealize the dispersion of glass using the index at d-light (.5875618 micrometers), the Abbe number, and a term which describes the deviation of the partial dispersion from the "Normal Line". The index at d-light is given the symbol Nd. The Abbe number (also called the V-number) is given the symbol Vd and is defined by

$$V_d = \frac{N_d - 1}{N_F - N_C},$$

where N_F and N_C are the indices of refraction at 0.4861327 and 0.6562725 micrometers, respectively. The partial dispersion term is $\Delta P_{g,F}$ .Zemax uses a formula based upon the typical dispersion of standard glasses in the visible range to estimate the index at any defined wavelength within the visible range as a function of the N_d and V_d values. This formula is accurate to roughly 0.0001 for typical glasses.

The N_d, V_d, and $\Delta P_{g,F}$ values are specified on the glass solve dialog box, which can be reached from the Lens Data Editor.

See also Using model glasses|topic=Using model glasses and Optimizing glass selection in the "Optimization Overview" reference file.

Note model glasses are an approximation, although usually a very good approximation in the visible range. Outside the visible wavelength range, such as in the ultraviolet or infrared, the model glass is not accurate and should not be used.

Polarization Analysis

Ray tracing programs generally treat rays as purely geometric entities, which have only a position, orientation, and phase. For example, a ray is completely described at a surface by the ray intercept coordinates, the direction cosines which define the angles the ray makes with respect to the local coordinate axes, and the phase, which determines the optical path length or difference along the ray.

At the boundary between two media, such as glass and air, refraction occurs according to Snell's law. Usually, the effects at the interface which do not affect beam direction are ignored. These effects include amplitude and phase variations of the electric field which depend upon the angle of incidence, the incident polarization, and the properties of the two media and any optical coatings at the interface.

Polarization analysis is an extension to conventional ray tracing which considers the effects that optical coatings and reflection and absorption losses have on the propagation of light through an optical system.

Review of polarization concepts

The Zemax User's Guide is not intended to be a tutorial on polarization theory. The subject is extensive and is treated well elsewhere. See for example "Polarization" by Jean M. Bennett, and "Optical Properties of Films and Coatings" by John A. Dobrowski, both in The Handbook of Optics Volume I, McGraw Hill, 1995.

The electric field vector

The amplitude and polarization state of the electric field is described by a vector:

$$E = \begin{bmatrix} E_x \\ E_y \\ E_z \end{bmatrix},$$

where Ex, Ey, and Ez are complex valued. The electric field vector must be orthogonal to the propagation vector of the ray. At a boundary between two media, the transmittance, reflectance, and phase of the electric field is different for the S and P components of the field. The S component of the field is the projection of E that lies along the axis orthogonal to the plane of incidence, while the P projection lies within the plane of incidence. The plane of incidence contains both the ray propagation vector and the surface normal vector at the intercept point. The electric field is then divided into Es and Ep components, both of which are complex valued.

The electric field after the boundary is computed by the consideration of the S and P transmittance (if the surface is refractive) or reflectance (if the surface is reflective). Zemax computes the reflectance and transmittance coefficients, but only uses the one appropriate to propagation of the ray. For brevity, the rest of this discussion will use the terms transmittance and refraction to indicate the terms of interest. If the surface is a reflector (i.e. the material type is MIRROR) then the reflectance coefficients will be used by the program automatically. The electric field after refraction is then

$$E_s' \,=\, E_s \tau_s, E_p' = E_p \tau_p \,,$$

where the transmittance coefficients, τ_s and τ_p, are complex valued. After the Es and Ep projections are computed, they are recomposed into the Ex, Ey, and Ez electric field vector and propagation continues to the next surface. The computation of the transmittance coefficients depends upon the index of refraction of the incident media, the (possibly complex) index of refraction and thickness of each layer in the optical coating (if any), and the (possibly complex) index of refraction of the substrate. The details of this computation are provided in The Handbook of Optics Volume I, McGraw Hill, 1995, and will not be repeated here.

Field vs. ray phase conventions

The thin film industry uses a different phase convention than is required for ray tracing. The thin film convention is to measure the phase shift along the normal vector as an imaginary plane wave propagates from the outermost coating layer to the substrate. This convention implies the phase shift is largest at normal incidence, and tapers off to zero by approximately the cosine of the angle of incidence for larger angles of incidence. Zemax calls the τ_s and τ_p values "Field" coefficients when using this reference.

For ray tracing, the optical phase advance or delay is measured along the ray. Zemax traces rays directly to the substrate, ignoring the coating thickness. This is because the thickness between surfaces specified in the Lens Data Editor is the thickness between the substrates, ignoring coatings. The coating is presumed to grow into the space preceding the surfaces. Correctly computing the phase of the ray requires back propagating the electric field to the point the coating begins, and adjusting the coating phase to be measured along the ray vector rather than the normal vector. Zemax the τ_s and τ_p values "Ray" coefficients when using this convention. A similar computation yields the factors for reflective surfaces, or for surfaces where the coating stack has automatically been reversed, and Zemax automatically chooses the correct factors in all cases.

The default in Zemax is to convert the Field coefficients to Ray coefficients, and this setting is recommended in all cases. However, the conversion may be disabled, see "Convert thin film phase to ray equivalent" in the Polarization section of the System Explorer.

The polarization ellipse

A convenient way of describing the polarization state of the electric field for rays parallel to the local Z axis only is to ignore the Ez component and consider only Ex and Ey. By plotting on a cartesian graph the endpoint of the vector (Ex, Ey) as time moves through one period, a figure is traced out which may be a straight line, a circle, or most generally an ellipse. This curve is defined by the major and minor axis lengths, and the angle the major axis makes with respect to the X axis on the cartesian graph. Note the minor axis may have a length of zero, in the case of linear polarization. For circular polarization the major and minor axis lengths are identical.

Whenever the polarization ellipse is used, it is important to remember that the Ez component is ignored; which limits the utility of the polarization ellipse in beams that are not nearly collimated along the local Z axis.

Definition of polarization terms

There are a number of ways to define parameters of interest for describing polarized light beams. All of the expressions generally depend upon angle of incidence, wavelength, and polarization orientation. Here are the terms used frequently by Zemax and their definitions:

Intensity, I

The intensity is the sum of the squares of the individual components of the electric field, or equivalently, the dot product of the electric field and it's complex conjugate:

$$I = E \bullet \tilde{E},$$

where the symbol E˜ denotes the complex conjugate of the vector E . Subscripts are applied to the symbol I to indicate which intensity is listed, for example, Ix would be the intensity in the x direction only.

Phase, P

The phase of any one component of the electric field can be computed with the arg function, which is the arctan(imaginary / real). The phase is given by

$$P = \arg(E) .$$

Subscripts are applied to the symbol P to indicate which intensity is listed, for example, Ps would be the phase for the S component of the electric field.

Path length through medium, τ

For homogeneous medium, the path length is the distance in lens units along the ray from the previous surface to the current surface. Note the index is accounted for separately; this is not the "optical" path length. For gradient index medium, the effective path length (for purposes of computing internal absorption) is the actual integral of the optical path length divided by the base index of the medium (the index at x=0, y=0, z=0). The path length is only used for computing the internal absorption. The path length will be negative for virtual propagations.

Internal absorption per lens unit, α

This factor accounts for the internal or bulk absorption propagating through glass. The absorption is computed from the data provided in the glass catalog. See [Defining Transmission Data](#) in the reference file "Using Glass Catalogs".

Internal Transmittance, IT

The internal transmittance is computed using the path length τ and internal absorption α:

$$ IT = e^{-\alpha\tau}. $$

Propagation Phase Factors, pc, ps

As the ray propagates from surface to surface, the electric field rotates according to

$$ e^{i\theta}, \theta = (-2\pi\tau)/\lambda , $$

where τ is the path length and λ is measured in the medium. The propagation factors are written as the cosine and sine of the angle θ. The "E Field After" from the previous surface will be multiplied by these factors to yield the input electric field to the current surface. Zemax adopts the negative sign convention for the propagation phase factors to be consistent with the convention of negative imaginary index to indicate absorption in coating materials as discussed in "Defining coatings in Zemax".

Amplitude Reflectance, ρ

The reflectance amplitude is the complex valued coefficient of reflectance for the electric field. These are computed for both S and P polarizations and using both field and ray phase conventions, as described above.

Amplitude Transmittance, τ

The transmittance amplitude is the complex valued coefficient of transmittance for the electric field. These are computed for both S and P polarizations and using both field and ray phase conventions, as described above.

Intensity Reflectance, Rs, Rp

Reflectance intensity is measured normal to the surface. It is given the symbol R, and is always real valued between 0 and 1. Reflectance can be computed from the reflectance amplitude as follows:

$$R = \rho^* \rho \,.$$

Intensity Transmittance, Ts, Tp

Transmittance intensity is measured normal to the surface. It is given the symbol T, and is always real valued between 0 and 1. Transmittance can be computed from the transmittance amplitude as follows:

$$T = \tau^* \tau \,.$$

Intensity Absorption, As, Ap

Absorption intensity is the intensity of the electric field which is neither transmitted nor reflected:

$$A = 1.0 - T - R \,.$$

Diattenuation, D

Diattenuation means "two attenuations", and is used to compare the loss of intensity of the S polarized light compared to the P polarized light. Diattenuation is defined as

$$D \,=\, \left| \frac{T_s - T_p}{T_s + T_p} \right| \,.$$

Coating Phase, Ps, Pp

The phase of the transmitted beam is generally different for the S and P polarizations. The phase is given by

$$P_s \,=\, \arg\,(\,E_s\,),\, P_p = \arg\,(\,E_p\,)\,.$$

Retardance, S

For transmission, the retardance is the phase of the P polarization minus the phase of the S polarization:

$$S = \arg(E_p) - \arg(E_s).$$

For reflection, the retardance is the phase of the P polarization minus the phase of the S polarization plus π:

$$S = \arg(E_p) - \arg(E_s) + \pi.$$

The different conventions are used to agree with common practice in the thin films industry.

Phase Difference Between X and Y, Pxy

This quantity is the phase difference between the X and Y components of the electric field, or

$$Pxy = Px - Py.$$

Note this value ignores the Z component of the electric field.

Major and Minor semi axis of the Polarization Ellipse, EM, Em

An ellipse has a major axis and a minor axis; the major axis being the longer of the two. These magnitudes, along with the angle of the polarization ellipse (Ap) uniquely define the polarization state in the XY plane.

Angle of XY Polarization Ellipse, Ap

This parameter is the angle relative to the local +x axis that the major axis of the polarization ellipse makes. If the polarization ellipse is a circle, this angle is meaningless.

Properties of uncoated surfaces

At normal incidence only, uncoated surfaces between dielectric media have transmittance and reflectance amplitudes which are described by the Fresnel expressions:

$$R = \left[\frac{n_1 - n_2}{n_1 + n_2}\right]^2, \text{ and } T = \frac{4 n_1 n_2}{[n_1 + n_2]^2}.$$

At angles other than normal incidence, the relations are more complex. For example, a surface between air and BK7 at a wavelength of 0.55 micrometers has reflectance and transmittance as a function of incident angle as shown in the following figures.
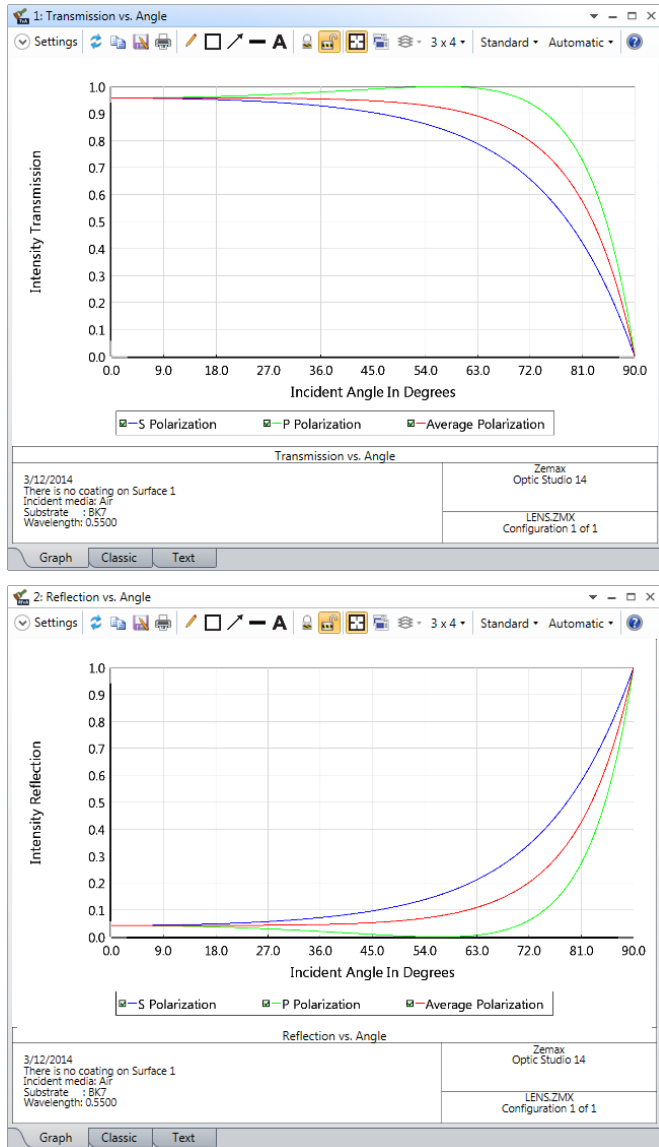
Note that the reflectance increases substantially with incident angle, until nearly all of the light is reflected rather than transmitted at grazing incidence angles. Note also that the S and P polarization states have different transmittance. This leads to a polarization dependent apodization of the aperture.

Defining coatings in Zemax

For dielectric materials, the index of refraction is purely real, and therefore the imaginary part of the index is zero. For metals, the index of refraction is complex. There are two sign conventions in common use for the imaginary part of the index of refraction. Zemax uses the following convention:

$$\eta = n + ik ,$$

where n is the usual index of refraction and k is the extinction coefficient, a negative value for absorbing materials. The alternate sign convention is $\eta = n - ik$ where k is then a positive value for absorbing materials, but Zemax does not use this convention. For example, using the Zemax convention, the index of aluminum is approximately given by

$$\eta = 0.7 - 7.0i .$$

Note the extinction coefficient is negative using this convention for typical absorbing materials. This choice of index sign convention is related to the sign convention used by the propagation phase factors, see "Propagation Phase Factors, pc, ps".

Zemax uses a text file format to define all coating data. The file may be of any valid file name. A sample file called COATING.DAT is supplied with Zemax, and COATING.DAT is the default coating file name for any new lens. Multiple coating files may be defined, and the name of the coating file use by any particular lens file is defined in the Files section of the System Explorer. It is highly recommended that any modifications to COATING.DAT be saved in a file with a different name, so that subsequent updates to Zemax will not overwrite the changes made to COATING.DAT.

In the coating file, the keywords MATE (for material), TAPR (for taper profile), COAT (for coating), TABLE (for coatings defined by a table of points) and IDEAL or IDEAL2 (for ideal coatings) are used to define different types of coating data. All of the material and taper definitions come first, then all of the coating definitions.

Coating file data syntax

The coating file consists of sections of data defining materials, tapers, and coatings. The data uses the following format and syntax. Each data section is described in detail in the text which follows.

```
! Any line starting with the ! symbol is a comment line
! For information on the MATE format see "The MATE data section" below.
MATE <material name>
wavelength real imaginary
wavelength real imaginary
....

MATE <next material name>
wavelength real imaginary
wavelength real imaginary
```

```
....

! For information on the TAPR format see "The TAPR data section" below.
TAPR <taper name>
DX decenterx
DY decentery
AN rotation_angle (degrees)
RT termnumber radialterm
CT termnumber cosineterm
PT termnumber polyterm
....

TAPR <next taper name>
DX decenterx
DY decentery
AN rotation_angle (degrees)
RT termnumber radialterm
CT termnumber cosineterm
PT termnumber polyterm
....

! For information on the COAT format see "The COAT data section" below.
COAT <coating name>
material thickness is_absolute loop_index tapername
material thickness is_absolute loop_index tapername
...

COAT <next coating name>
material thickness is_absolute loop_index tapername
material thickness is_absolute loop_index tapername
...

COAT I.transmission

ENCRYPTED filename

! For information on the IDEAL format see "The IDEAL data section" below.
IDEAL <name> <Transmitted intensity> <Reflected intensity>

! For information on the IDEAL2 format see "The IDEAL2 data section" below.
IDEAL2 <name> s_rr s_ri s_tr s_ti p_rr p_ri p_tr p_ti no_pi_flag

! For information on the TABLE format see "The TABLE data section" below.
TABLE <coating name>
ANGL <angle in degrees>
WAVE <wavelength 1 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 2 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 3 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
...
ANGL <next angle in degrees>
WAVE <wavelength 1 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 2 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 3 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
...
```

For details on each of these coating definition keywords, see the discussions which follow.

The MATE data section

Syntax:

```
MATE <material name>
wavelength real imaginary
wavelength real imaginary
....
```

The material names may be any user defined name up to 20 characters in length, with no special characters permitted.

Zemax uses the material data in the following way:

The wavelength argument is always in micrometers. The wavelengths must be specified in ascending order.

The real value is the real index of refraction of the material at that wavelength.

The imaginary value is the extinction coefficient.

If a single wavelength value is provided for a material, then the real and imaginary parts of the index of refraction are used no matter what wavelength is being traced. Dispersion of the coating material is therefore ignored.

If two or more wavelengths are defined for a material, then for wavelengths shorter than the shortest defined wavelength, the shortest wavelength data is used. For wavelengths longer than the longest defined wave- length, the longest wavelength data is used. For wavelengths in between, linear interpolation is used.

The TAPR data section

Syntax:

```
TAPR <taper name>
DX decenterx
DY decentery
AN rotation_angle (degrees)
RT termnumber radialterm
CT termnumber cosineterm
PT termnumber polyterm
....
```

Normally, Zemax assumes the coating thickness is uniform across the entire optical surface. Because of the way some optical coatings are applied, a thin film coating may have a thickness that varies over the optical surface. A tapered coating is defined by applying a taper function which computes a dimensionless coefficient to multiply the nominal coating thickness. Any layer of any coating may have a taper applied, and each layer may have a different taper. The

form of the taper may be a radial polynomial, a cosine function, or a general polynomial in the x and y directions. The various models may all be used simultaneously, and the taper function may optionally be decentered and rotated.

The effective coating thickness is:

$$d_{eff} = d \times (f_r + f_c + f_p),$$

where d is the nominal coating thickness and fr, fc, and fp are the radial, cosine, and polynomial taper factors, respectively. The taper factors are functions of the decentered and rotated coordinates (x', y'), which are computed from the local surface coordinates using these expressions:

$$x' = (x - x_d)\cos\theta - (y - y_d)\sin\theta,$$

$$y' = (x - x_d)\sin\theta + (y - y_d)\cos\theta,$$

where θ is the taper function rotation angle specified in degrees. The radial coordinate is then defined as:

$$r = \sqrt{(x')^2 + (y')^2} \quad .$$

The radial taper factor is defined as:

$$f_r = \sum_{i=0}^{22} \beta_i r^i,$$

The cosine taper factor is defined as:

$$f_c = \sum_{i=0}^{5} \delta_i (\cos\alpha)^i,$$

where α is defined as:

$$\sin\alpha = \left|\frac{r}{R}\right|,$$

where r is the radial coordinate as defined above. For coatings applied to sequential surfaces, R is the radius of curvature of the surface as defined by the "Radius" column in the Lens Data Editor. Cosine tapers should only be used on radially symmetric sequential surfaces for this reason. For coatings applied to non-sequential objects, R is always infinity, regardless of the shape of the object, and cosine tapers should not be used.

The polynomial taper factor is defined as:

$$f_p = \sum_{i=0}^{22} \gamma_i E_i,$$

where E_i are the polynomial terms in x' and y'. The E_i terms are a power series:

$$E_0 = 1, \; E_1 = x', \; E_2 = y', \; E_3 = (x')^2, \; E_4 = x'y', \; E_5 = (y')^2, \; E_6 = (x')^3, \text{ etc.}$$

Note the cosine taper factor will yield different coatings on substrates that have a different radius of curvature. To define a coating whose thickness varies exactly as $\cos \theta$, the δ_0 cosine taper factor should be zero and the δ_1 cosine taper factor should be 1.0. When applying cosine taper factors to coatings on non-sequential component objects; the substrate radius is ignored and treated as a plane. The radial and polynomial terms are still used.

If the sum of the taper factors evaluates to a negative number, Zemax assumes it has a value of zero. The parameters x_d, y_d, θ, γ_i, δ_i, and β_i are all defined in the taper data section of the coating file. Note that both even and odd powers of the radial coordinate are used, and that the zero order term is used. The coefficients generally have units and magnitudes that depend upon the lens units.

> **Care should be taken to check that the proper lens units are being used when defining or using tapered coatings.**

The taper names may be any user defined name up to 20 characters in length, with no special characters permitted.

Zemax uses the taper data in the following way:

If the first keyword on any line following the TAPR keyword is DX or DY, then the data value after the DX or DY keyword on that same line is assumed to be the x_d or y_d in lens units, respectively.

If the first keyword on any line following the TAPR keyword is AN, then the data value after the AN keyword on that same line is assumed to be the taper rotation angle θ.

If the first keyword is RT, the second data value must be an integer between 0 and 20, inclusively. This integer determines the radial term number. The third data value is the coefficient on that term, β_i.

If the first keyword is CT, the second data value must be an integer between 0 and 5, inclusively. This integer determines the cosine term number. The third data value is the coefficient on that term, δ_i.

If the first keyword is PT, the second data value must be an integer between 0 and 20, inclusively. This integer determines the polynomial term number. The third data value is the coefficient on that term, γ_i.

Any terms Zemax uses are assumed to be zero unless otherwise defined.

If the entire layer of a coating is to be scaled by a different amount on each surface, or if the coating thickness needs to be optimized, see the discussion about coating multipliers in the "Coating" section of "Surface Properties" in "The Setup Tab" help file, and the "Optimizing coatings with Zemax" section in this reference file.

The COAT data section

The order of the coating layers is important!

Syntax:

```
COAT <coating name>
material thickness is_absolute loop_index tapername
material thickness is_absolute loop_index tapername
...
```

or

```
COAT I.transmission
```

The coating names may be any user defined name up to 20 characters in length, with no special characters permitted.

Zemax uses the coating data in the following way:

When the coating file is first read, Zemax verifies that each coating consists of materials that were defined in the material section. If referenced materials are not defined, an error is issued.

The coating thickness is measured either in units of primary wavelength thickness in that medium (relative definition), or in units of micrometers independent of any wavelength (absolute definition). If relative definition is used, the same coating used by two different lens files, with two different primary wavelengths, will have different actual coating thicknesses. The actual thickness of the coating is determined by:

$$d = \frac{\lambda_0}{n_0} T,$$

where λ_0 is the primary wavelength in micrometers, n_0 is the real part of the index of refraction of the coating at the primary wavelength, and T is the "optical thickness" of the coating specified in the coating file. For example, in a coating material whose real part of the index of refraction is 1.4, a quarter wave coating (T = 0.25) at a primary wavelength of 0.550 micrometers would be 0.0982 micrometers thick. Note that only the real part of the index of refraction is used for determining the thickness of the layer.

If the is_absolute integer is zero, the thickness is interpreted to be relative, otherwise, the coating thickness is absolute in micrometers. If the is_absolute, loop_index, and tapername are omitted, then is_absolute and loop_index are both assumed to be zero, and no taper is assumed.

The loop_index parameter determines how Zemax replicates groups of coating layers. See the next section for details.

Defining replicated groups of coating layers

Some coatings contain repeated groups of coating layers. For example, a coating definition may be defined as follows:

```
COAT 3GROUPS
MAT0 0.25 0 0
MAT1 0.25 0 0
MAT2 0.50 0 0
MAT3 0.25 0 0
MAT1 0.25 0 0
MAT2 0.50 0 0
MAT3 0.25 0 0
MAT1 0.25 0 0
MAT2 0.50 0 0
MAT3 0.25 0 0
MAT4 0.25 0 0
```

Note the sequence of layers with materials MAT1, MAT2, and MAT3 is repeated 3 times. This is a perfectly acceptable syntax, however, 11 text lines and 11 layers are required to define the coating. A shorthand syntax that replicates the layers is available using the loop_index parameter. The loop_index integer parameter is set to the number of times a group of layers is to be repeated. The loop_index must be set at BOTH the first AND the last layer (this is to indicate the range of layers, and Zemax also reads the coating data in both directions, as described in a following section). The coating above could be written as:

```
COAT 1GROUP
MAT0 0.25 0 0
MAT1 0.25 0 3
MAT2 0.50 0 0
MAT3 0.25 0 3
MAT4 0.25 0 0
```

Note the "3" parameter appears on both the first and third layers listed. This syntax not only saves typing, it reduces the chance of careless typing errors, eases editing, and conserves the total number of layers; which is limited for each coating definition (see the section "Limits on the amount of coating data" later in this section). The coating above only requires 5 "layers" to define, although there are physically 11 layers modeled.

There are some important points regarding the use of the loop_index:

There are no limits on the number of different groups that may be defined, or on how many layers each group contains, as long as the total number of lines under the COAT heading does not exceed the maximum number of allowed layers, defined below.

A loop_index of zero is the same as a loop_index of 1; that is, zero does not mean do not include a group at all!

Loops may not be nested, nor may they overlap.

Defining simple ideal coatings

The COAT keyword also may be used to define simple ideal coatings. The syntax is

```
I.transmission
```

where transmission is numerical data. For example, the command

```
COAT I.75
```

will create a coating named I.75 which transmits 75% of the ray energy. Ideal coatings defined in this way transmit the specified fraction of light, and reflect the rest, independent of the ray wavelength, angle of incidence, incident material, or substrate material. Note this type of ideal coating has no absorption, and the transmission must be less than 1.0 and greater than 0.0. If the ray total internal reflects, the reflected intensity factor is assumed to be 1.0. For a more general ideal coating, see the IDEAL and IDEAL2 coating definitions which follow.

The ENCRYPTED data section

Syntax:

```
ENCRYPTED filename
```

Encrypted coating files are created when a coating is exported to a Zemax Encrypted Coating file format as described in "Export Encrypted Coating" in the "Coating Tools" section of "The Libraries Tab" help file. The resulting ZEC file must be placed in the coating file folder. The filename should have no path or extension; the extension ZEC will automatically be appended.

Encrypted coatings and material data are not listed in the coating listing or prescription reports.

The IDEAL data section

Syntax:

```
IDEAL <name> T R TIR
```

Ideal coatings are defined only by three intensity coefficients: transmission, reflection, and total internal reflection. Ideal coating names may be any user defined name up to 20 characters in length, with no special characters permitted. After the ideal coating name, the T, R, and TIR values are listed. If T+R exceeds 1.0, the values will be scaled so that T+R = 1.0. The absorption coefficient is computed automatically via A = 1.0 - R - T, to conserve energy. To make a coating that absorbs 100% of the energy, set both T and R to zero. Zemax assumes the phase of transmission is zero. For mirror substrates, the phase of reflection is π. For other substrates, the phase of reflection is zero if the incident index is higher than the substrate index, otherwise the phase of reflection is π. The TIR intensity is used only if the ray total internal reflects. If the TIR value is omitted, 1.0 is assumed.

The IDEAL2 data section

Syntax:

```
IDEAL2 <name> s_rr s_ri s_tr s_ti p_rr p_ri p_tr p_ti no_pi_flag TIR
```

The Ideal2 keyword provides an alternate means of defining ideal coatings. Ideal2 coatings are defined by the real and imaginary amplitude transmission and reflection coefficients for S and P polarized light separately. Ideal2 coating names may be any user defined name up to 20 characters in length, with no special characters permitted. After the coating name, the reflection real, reflection imaginary, transmission real, and transmission imaginary values are listed for S polarized light, followed by these same four values for P polarized light. To make a coating that absorbs 100% of the energy, set both transmission and reflection values to zero. The phase of reflection and transmission is defined by the complex coefficients. If the substrate is not a mirror, and the incident index is higher than the substrate index, Zemax changes the sign of the reflected real and imaginary coefficients to yield an additional phase change upon reflection of π, unless the no_pi_flag is set to any value other than zero; in which case no changes are made to the phase of the coating. The TIR intensity is used only if the ray total internal reflects. If the TIR value is omitted, 1.0 is assumed.

The TABLE data section

Syntax:

```
TABLE <coating name>
ANGL <angle in degrees>
WAVE <wavelength 1 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 2 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 3 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
...
ANGL <next angle in degrees>
WAVE <wavelength 1 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 2 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
WAVE <wavelength 3 in micrometers> Rs Rp Ts Tp Ars Arp Ats Atp
...
```

Table coatings are similar to IDEAL coatings, except the transmission and reflection may be a function of incident angle and wavelength and may be specified separately for S and P polarizations. For multiple discrete angles of incidence and wavelengths, four intensity values are defined: Rs, Rp, Ts, and Tp; for intensity reflection and transmission and the s and p subscripts indicate the polarization state. There are also four angles (in degrees) defined: Ars, Arp, Ats, and Atp; these values define the phase angle of the R and T values so Zemax can compute the real and imaginary parts of the transmission and reflection. The real part is proportional to the cosine of the angle, and the imaginary part is proportional to the sine of the angle. If the angles are omitted or zero the reflection or transmission is real and no phase change is introduced by the coating. The absorption is computed from A = 1 - R - T for each polarization state. If the substrate is not a mirror and the incident index is higher than the substrate index, the phase of reflection is changed by a factor of π .

The primary purpose of the table model is to allow accurate ray tracing where the layer by layer prescription of the coating is not known or is unavailable because the coating vendor will not provide the design data.

The syntax consists of the keyword TABLE followed by the name of the coating. After the TABLE keyword, blocks of data are defined for each incident angle. After each angle definition, R and T values are defined for each wavelength in the table. An example syntax would be:

```
TABLE MYTABLECOAT
ANGL 0
WAVE 0.35 0.014 0.014 0.986 0.986 0.0 0.0 0.0 0.0
WAVE 0.70 0.013 0.013 0.987 0.987 0.0 0.0 0.0 0.0
ANGL 5
WAVE 0.35 0.015 0.015 0.985 0.985 0.0 0.0 0.0 0.0
WAVE 0.70 0.012 0.012 0.988 0.988 0.0 0.0 0.0 0.0
ANGL 45
WAVE 0.35 0.065 0.070 0.935 0.930 0.0 0.0 0.0 0.0
WAVE 0.70 0.062 0.065 0.938 0.935 0.0 0.0 0.0 0.0
ANGL 80
WAVE 0.35 0.600 0.600 0.400 0.400 0.0 0.0 0.0 0.0
WAVE 0.70 0.600 0.600 0.400 0.400 0.0 0.0 0.0 0.0
```

There are several very important points to note:

Angles of incidence less than the smallest angle defined will use the smallest angle data. Angles of incidence greater than the largest angle defined will use the largest angle data. Other values are linearly interpolated in cosine of the angle space. Note Zemax interpolates but does not extrapolate. Angles must be listed in ascending order. If data for an angle of incidence of 0.0 degrees is defined, the S and P data should be identical, since S and P are indistinguishable at normal incidence.

Angles of incidence are assumed to be in the incident media, and for coatings placed on refractive surfaces, that the incident media has a lower index of refraction than the substrate. For coatings placed on surfaces where the rays go from high index to low index (such as glass to air), the refracted angle in the low index media is used as the angle of incidence. Zemax assumes the phase of transmission is zero if no phase angles are defined. The phase data is angles measured in degrees if the data is provided. If the substrate is not a mirror and the incident index is higher than the substrate index, the phase of reflection is changed by a factor of π .

Wavelengths in between wavelength data points will linearly interpolate in wavelength space.

All wavelengths listed must be identical for each angle. The same ascending order values for the wavelengths is required at each angle for accurate interpolation. An error message will result if the wavelengths are not properly defined.

Generally much more than the brief set of data listed here is required for adequate accuracy. There are no limits on the number of angle and wavelength data points associated with each table coating.

Adding comments to the coating file

Any text following the "!" symbol is considered a comment.

Limits on the amount of coating data

The number of different coating names, materials, table coatings, and other data are in general only by available memory. There are some limits to data that may be stored in any one catalog as follows:

400 layers per coating (see the section "Defining replicated groups of coating layers" earlier in this file to work around this limit if required).

120 dispersion data points per material.

60 different tapers.

One way to work around these limits is to use a different coating file for each lens, if required. These limits only apply to a single coating file; multiple coating files may be used for different lenses.

Editing the coating file

There are a few ways of accessing the data in the coating file. First, the file can be edited outside of Zemax using any text editor. Second, there is a "Edit Coating File" option under the "Tools" menu on the main menu bar. This invokes the Windows NOTEPAD editor. Also, there is a "Coating Listing" option under the "Tools" menu, which lists the coating and material definition data. If the coating file is edited, the file must be reloaded in order to update the new coating data; there is a menu option on the Tools menu for reloading the catalog.

Optimizing coatings with Zemax

Zemax is not intended to be an optical coating design program. Optimization of coating designs generally requires some automation of the coating material selection and number of layers.

However, Zemax does have the ability to optimize and tolerance coating layer thicknesses, indices, and extinction coefficients, once the number of layers and material selection is made. Zemax uses a dimensionless "coating multiplier"

which linearly scales the thickness of any layer of a coating, and dimensionless offsets of the index and extinction coefficients. Optimization may then be made on the usual coating performance data using the CODA operand. To optimize the layer thicknesses or offsets, first define a coating with the correct number of layers and materials for each layer. The initial coating thicknesses should be set to either a known good starting value, or 1 wave (in this latter case the coating multipliers may be interpreted to be in units of waves). Do not use zero for a coating layer thickness as this thickness cannot be linearly scaled. Coating multipliers may not exceed the value 10.0. The initial material index and extinction should be set to 1 and 0, respectively, or to a known good starting value.

Coating multipliers and index offsets may then be defined on the surface coating tab described in the "Coating" section of "Surface Properties" in "The Setup Tab" help file. This feature supports setting the multipliers and index offsets to a variable status, which allows for optimization of the relative coating layer thickness and index of refraction of the coating materials. To constrain the coating multiplier values, see "CMGT" and the related operands "CMLT" and "CMVA" in the "Optimization Overview" reference file. To constrain the index and extinction offsets, see "CIGT" and the related operands "CILT", "CIVA", "CEGT", "CELT", and "CEVA" in the "Optimization Overview" reference file.

Default materials and coatings supplied with Zemax

The default COATING.DAT file supplied with Zemax contains several materials and a few common optical coatings. The following table describes some of the included materials and coatings.

DEFAULT MATERIALS AND COATINGS

Material	Description
AIR	Unity index, used for including air gaps in coatings
AL2O3	Aluminum Oxide, index 1.59
ALUM	Aluminum, index 0.7-7.0i
ALUM2	An alternate definition for aluminum
CEF3	Cerous Flouride, index 1.63
LA2O3	Lanthanum Oxide, index 1.95
MGF2	Magnesium Fluoride, index 1.38
N15	Imaginary material of index 1.50
THF4	Thorium Fluoride, index 1.52
ZNS	Zinc Sulphide, index 2.35
ZRO2	Zirconium Oxide, index 2.1

Coating	Description
AR	General anti reflection, defined as a quarter wave of MGF2
GAP	A small gap of air used to show evanescent propagation

HEAR1	High performance anti reflection coating
HEAR2	High performance anti reflection coating
METAL	A thin layer of aluminum used to make beamsplitters
NULL	A zero thickness coating used primarily for debugging
WAR	Anti-reflection "W" coat, defined as a half wave of LA2O3 followed by a quarter wave of MGF2.

These materials and coatings are meant to be used as examples, and may not be applicable in any particular situation. Always check with coating manufacturer or designer for details on coating material specifics. For more detailed information on how these materials and coatings are defined, use the Coating/Material Listing report described in the "Coating Catalog" section of "The Libraries Tab" help file.

Specifying coatings on surfaces

Once a coating has been defined, it can be applied to a surface by specifying the coating name in the coating column; located at the extreme right side of the lens data editor. Zemax interprets the coating definition using one of four rules:

If the surface specified is a boundary going from air to glass, the coating layer order is interpreted exactly as specified in the coating file. The incident media is air, then the outermost layer is listed first (at the top) of the coating definition, then the next layer, etc., with the substrate being the glass type on the surface. The definition of the coating should not include the substrate index or material definition. The term glass here means the glass type is not "MIRROR", and not blank (which is treated as unity index for air). Gradient index lenses are considered glass.

If the surface specified is a boundary between air and air, or glass and glass, the coating is also interpreted exactly as for air to glass, with the appropriate calculations done for the incident and substrate media.

If the surface specified is a boundary going from glass to air, then the order of the layers is automatically reversed, so that the coating is the same as if it had been applied going from air to glass. Therefore, a coating defined by ALHHS, would be interpreted as SHHLA if the boundary goes from glass to air.

If the surface type is a mirror, then the coating definition must include the substrate index. The last layer in the coating definition is then assumed to be a semi-infinite thickness of substrate material.

Zemax may also model certain limiting cases, such as frustrated TIR; see "Modeling frustrated total internal reflection" described later in this file.

What Zemax does if no coating is specified

If the coating column is left blank, then the following is assumed:

If the surface is a mirror, then the surface is assumed to be coated with a thick layer of aluminum, with an index of refraction 0.7 - 7.0i. The aluminum layer is assumed to be thick enough that no light propagates past the layer.

If the surface is a dielectric interface, bare, uncoated surfaces are assumed.

Defining the initial polarization

The individual analysis settings, such as those for the polarization ray trace (see Polarization Ray Trace) allow for specification of the input polarization. The polarization is defined using a Jones vector:

$$J = \begin{bmatrix} J_x \\ J_y \end{bmatrix},$$

where Jx and Jy have both a magnitude and a phase, and the symbol J is used instead of E to distinguish the 2D Jones vector from the 3D electric field vector E. Zemax normalizes the specified Jx and Jy values to have unity magnitude, and then scales the intensity as appropriate if any pupil apodization has been specified. The Jx and Jy values are therefore measured in terms of relative electric field amplitude.

Let the ray vector be K, which has X, Y, and Z direction cosines (l, m, n). For rays traveling parallel to the Z axis, or K = (0, 0, 1), the electric field in the Z direction is identically zero, and the conversion from the Jones vector to the Electric field is trivial:

$$E_x = J_x, E_y = J_y, E_z = 0.$$

For a more general ray, the conversion of the Jones vector (Jx, Jy) to the 3D electric field values (Ex, Ey, Ez) is ambiguous. It is not possible to interpret the Jx and Jy values as meaning that for any ray, the Jx value should be applied in such a way to leave the Ey zero, and for the Jy to be applied so that the Ex is zero. The reason is that the E resulting for the Jones vectors (Jx = 1, Jy = 0) and (Jx = 0, Jy = 1) must be orthogonal to both K and to each other.

Zemax allows user selection of three different methods to perform the conversion from J to E. In each method, the vector K refers to the ray vector, the Jx value is the field along a vector S, and the Jy value is the field along a vector P. Note that K, S, and P must all be unit vectors and orthogonal to each other. The three methods are:

X Axis Reference: The P vector is determined from K cross X, and S = P cross K. This method is the default.

Y Axis Reference: The S vector is determined from Y cross K, and P = K cross S.

Z Axis Reference: The S vector is determined from K cross Z, and P = K cross S.

When the object is at infinity, the method selected will change the polarization orientation of S and P for different fields, but all rays from the same field will have the same polarization since all rays are parallel to each other. For finite conjugates, especially when the object space numerical aperture is large, the S and P vector orientations will vary for different rays in the pupil. No matter which method is selected, the transmission results for unpolarized light will be unaffected because any two orthogonal rays may be traced to compute transmission. For systems where a particular polarization is desired, care should be taken to verify that the conversion from J to E yields the intended polarized ray. To review the detailed conversion results, see "Polarization Ray Trace" in the "Polarization" section of "The Analyze Tab" help file.

Defining polarizing components

Any boundary between two media can polarize a beam. However, Zemax supports an idealized model for a general polarizing device. The model is implemented as a special "Jones Matrix" surface type for sequential ray tracing, and a "Jones Matrix" object type for non-sequential ray tracing. The Jones matrix modifies a Jones vector (which describes the electric field) according to

$$\begin{bmatrix} E'_x \\ E'_y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix},$$

where A, B, C, D, Ex, and Ey are all complex numbers. In the lens data and in the non-sequential components editor, Zemax provides cells for defining A real, A imag, etc.

Note that what happens to the z component of the electric field is not defined by the Jones matrix surface, which implicitly assumes the ray is at normal incidence. The E component of the electric field must satisfy the boundary condition:

$$\vec{k} \bullet \vec{E} = 0,$$

where E is the electric field after the Jones matrix and k is the ray vector. Zemax will adjust either the Ez component or the Ex and Ey components, such that the magnitude of E does not increase. The adjustment may however require a reduction in the magnitude of E, and thus an associated loss of transmitted energy.

The Jones matrix can be used to define a wide variety of polarizing components. For example, see the sample Jones matrices in the following table. The format of the numbers is (a, b) where the "a" value is the real part and "b" is the imaginary part.

SAMPLE JONES MATRICES

Component	Matrix
Null matrix	$\begin{bmatrix} (1,0) & (0,0) \\ (0,0) & (1,0) \end{bmatrix}$
X analyzer	$\begin{bmatrix} (1,0) & (0,0) \\ (0,0) & (0,0) \end{bmatrix}$
Y analyzer	$\begin{bmatrix} (0,0) & (0,0) \\ (0,0) & (1,0) \end{bmatrix}$
Quarter-Wave plate in X direction	$\begin{bmatrix} (0,1) & (0,0) \\ (0,0) & (1,0) \end{bmatrix}$
Quarter-Wave plate in Y direction	$\begin{bmatrix} (1,0) & (0,0) \\ (0,0) & (0,1) \end{bmatrix}$
Half-Wave plate in X direction	$\begin{bmatrix} (-1,0) & (0,0) \\ (0,0) & (1,0) \end{bmatrix}$
Half-Wave plate in Y direction	$\begin{bmatrix} (1,0) & (0,0) \\ (0,0) & (-1,0) \end{bmatrix}$
Intensity filter, 25% transmission	$\begin{bmatrix} (0.5,0) & (0,0) \\ (0,0) & (0.5,0) \end{bmatrix}$

What Zemax can compute using polarization analysis

Zemax can generate plots of R, T, A, D, P, and S as a function of either wavelength for a given incidence angle or as a function of angle given the wavelength. See the "Image Quality" section of "The Analyze Tab" help file for details.

Using the "Polarization Ray Trace" feature, Zemax can compute and tabulate the detailed calculations used to trace the polarization state through the system for any given ray. Other features directly compute polarization dependent results; see the "Polarization and Surface Physics" section of "The Analyze Tab" help file for details.

Bulk absorption and transmission

Zemax also can accurately compute transmission of individual rays, or compute an average over the pupil. Transmission calculations are based upon surface effects, such as reflection losses, as well as bulk transmission through glass according to Beer's law:

$$t = e^{-\alpha\tau},$$

where α is the absorption coefficient and τ is the path length through the glass. The parameter α generally depends upon wavelength and has units of inverse length. See the "Defining Transmission Data" section of the "Using Glass Catalogs" reference file for details on defining transmission data. Polarization effects are not considered in all Zemax computations.

Modeling birefringent materials

Birefringent materials in sequential optical systems are modeled using the birefringent in/out surface types, described in the "Birefringent In and Birefringent Out" part of the "Sequential Surfaces" reference file. For non-sequential systems, see "Birefringent Types" in the "Index" section of the "Object Properties" in the NSC Editor.

Modeling frustrated total internal reflection

Frustrated TIR (FTIR) occurs when a ray of light traveling through glass strikes an interface at an angle exceeding the critical angle. If another dielectric medium, such as another piece of glass, is placed close, but not touching the interface, some light will transmit through the thin gap and propagate, even though the refraction at the boundary cannot satisfy Snell's law. Both the reflected and the transmitted beams will be affected, depending upon the thickness of the gap. In the limit of the gap having zero thickness, the light will continue through as if there were no boundary. In the limit of a large gap, more than a fraction of a wavelength, then virtually all of the light is perfectly reflected.

Although this appears to be a special case, the very general model incorporated into the polarization routines of Zemax handle FTIR as any other coating problem. The trick is to define a coating that reflects the situation. One of the default coatings is GAP, defined to be 0.1 waves of AIR, with index unity. Using the tilted surface with a Y-Tangent of 1.0, and placing the coating GAP on the boundary yields the following system, with correctly modeled polarization properties. Note the air gap may be changed by editing the coating file as described in the "Defining coatings in Zemax" section of this reference file.

With the GAP coating in place, Zemax can now compute either transmitted or reflected polarization properties.

The following layout is an example of a FTIR beamsplitter in transmission. The tilted surface between the two glass blocks has a GAP coating.

3D Layout
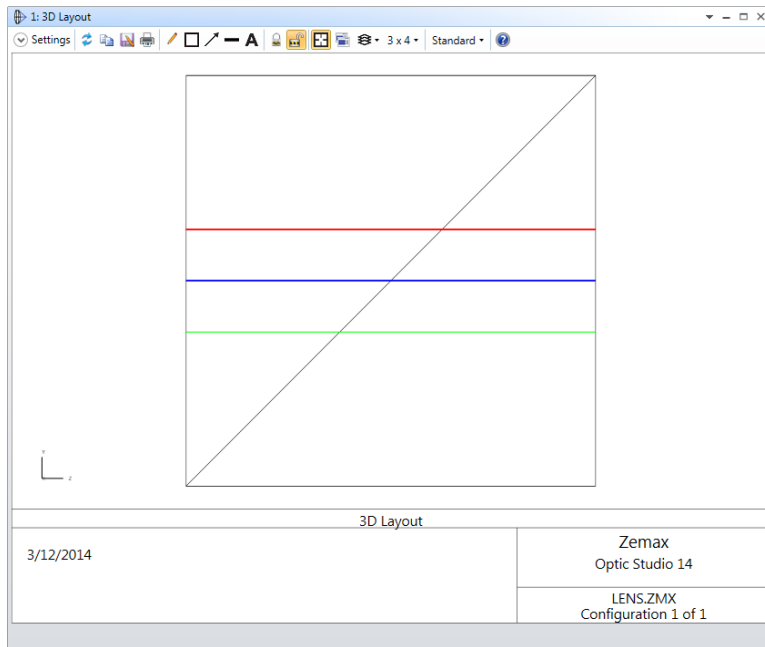
The graph below shows the transmission versus angle for the FTIR beamsplitter in transmission.

Transmission vs. Angle
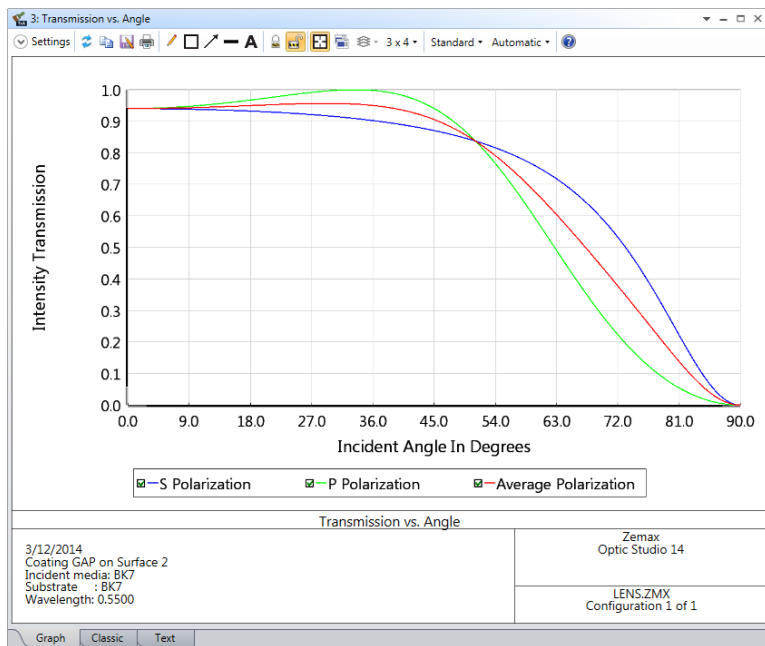
However, the ray tracing for other purposes (such as computing system transmission, ray fans, optimization) will only proceed along the transmitted path. To model the reflected path, give the internal tilted surface a glass type of MIRROR, add a 90 degree coordinate break after the tilted surface, and change the sign of the remaining thicknesses as usual.

The coating in this case now needs to be modified to indicate the presence of the second dielectric block, even though no ray tracing will be conducted in the transmitted path; just the reflected path. A new coating needs to be defined in the coating file (called FTIR in this example) that looks like:

```
COAT FTIR
AIR .1
N15 1
```

The "coating", when applied to a mirror, treats the mirror as a gap of air followed by piece of glass. Note that detailed modeling of the index and dispersion of the substrate glass is certainly possible by definition of a glass in the coating file; here N15 was used for simplicity. The thickness of the last material, N15, which defines the substrate, is irrelevant because Zemax always assumes the substrate is semi-infinite when applied to a MIRROR. The resulting lens figure and reflection plot is shown. Both the reflection and transmission plots are of course identical for either system model.

Finally, note that conventional TIR can also be modeled by increasing the gap thickness. Distances much greater than a fraction of a wavelength are essentially equivalent to an infinite gap. However, very large thicknesses for evanescent wave propagation should not be defined in the coating file because of "underflow" numerical errors which arise because the transmission is exceedingly low. One wave, or an optical thickness of 1.0, is usually sufficient.

The following layout is an example of a FTIR beamsplitter in reflection. The tilted "mirror" surface has a FTIR coating.
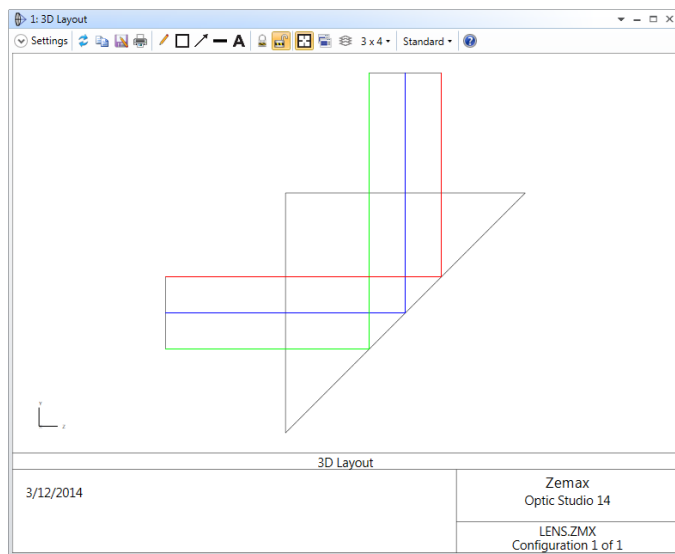
The graph below shows the reflection versus angle for the FTIR beamsplitter in reflection.
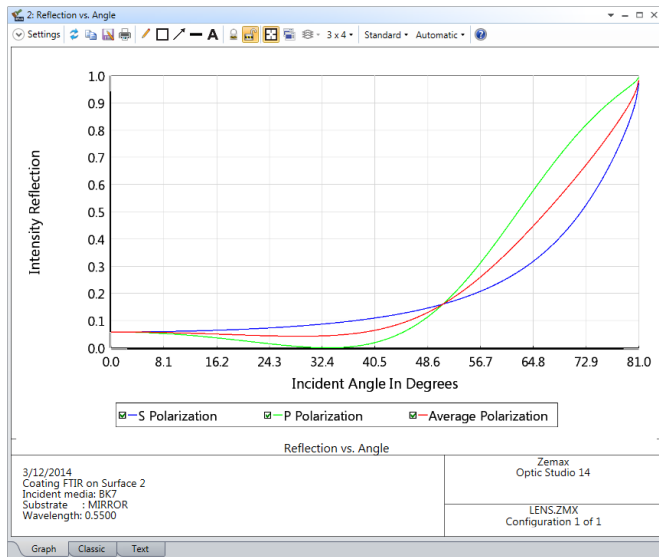
Limitations of polarization analysis

There are a few limitations to the polarization analysis capabilities of Zemax. The data in the coating file should always be carefully checked for accuracy whenever fabrication decisions are being made. Also, the reflectivity and transmission curves generated by Zemax should always be presented to the coating manufacturer as a check to ensure the coatings will behave as intended.

The polarization ray tracing algorithm generally does not return correct results for certain types of surfaces, such as paraxial surfaces, diffraction gratings, binary optics, or Fresnel surfaces. Generally, only surfaces which have refraction or reflection described by Snell's law have valid polarization data computed. In particular, the polarization algorithm assumes that rays remain in the plane of incidence after refraction or reflection. This is true for conventional refractive optics (it is a condition of Snell's law) but is not true in general for diffractive optics and "imaginary" surfaces such as paraxial lenses.

The polarization effects of propagation through gradient index media are not considered, although the surface effects (reflection, transmission coefficients) are accounted for. Because the rays take curved paths through gradient media, the polarization vector is rotated in a manner Zemax does not account for.

Optimization Overview

The Local Optimization (or simply, "Optimization") feature provided by Zemax is quite powerful, and is capable of improving lens designs given a reasonable starting point and a set of variable parameters. Variables can be curvatures, thicknesses, glasses, conics, parameter data, extra data, and any of the numeric multi-configuration data. Zemax uses either an actively damped least squares or an orthogonal descent algorithm. The algorithms are capable of optimizing a merit function composed of weighted target values; these target values are called "operands". Zemax has several different default merit functions, described in a subsequent section. These merit functions can be changed easily using the Merit Function Editor. For details on this procedure see "Modifying the merit function".

Optimization requires three steps: 1) a reasonable system which can be traced, 2) specification of the variables, and 3) a merit function. A reasonable system is a rather loose concept which means that poorly conceived designs are not likely to be transformed into exceptional designs by the optimization algorithm (although there are exceptions). The variables, and there must be at least one for the optimization algorithm to be able to make any progress, are specified on the various editors, as described in the next section.

The algorithm used by the optimization feature described in this chapter is designed to find the "local" minimum of the specified merit function. However, Zemax also has the capability to search for a "global" minimum of the merit function. The global minimum is the lowest possible value for the merit function, and if the merit function is selected appropriately, this implies the best possible solution to the problem. The global optimization feature is not for novice users, and is not appropriate for interactive designing. For details see "Global Optimization".

Selecting optimization variables

Variables for optimization are specified by pressing Ctrl-Z when the highlighted bar is on the parameter to be varied in the Lens Data Editor. See the "Lens Data Editor" section of The Setup Tab. Note that Ctrl-Z is a toggle. The Multi-Configuration Editor also contains numeric data that may be made variable by using Ctrl-Z. Glasses cannot be made variable directly because they are discrete. To optimize glasses, see the "Global Optimizers" section of The Optimize Tab.

Modifying the merit function

The merit function can be modified by the user. To change the merit function, open the Merit Function Editor from the Editors section of the Setup tab, or the Automatic Optimization section of the Optimize tab.. New operands can be added to the list, or others deleted, using the insert and delete keys. The current merit function value and the value of each operand can be updated by clicking the Update button. See the "Merit Function Editor Toolbar" section of The Optimize Tab.

Operands are set by typing the name in the first column and then filling in the remaining data fields. There are multiple fields that may be required to define an operand. The fields are called Int1 and Int2 for the two integer values, and Data1 through Data6 for up to 6 double precision values. Not all of the operands use all of the fields provided.

Hx, Hy, Px, and Py

Many of the operands use Data1 through Data4 for the values of Hx, Hy, Px, and Py; these are the normalized field and pupil coordinates (see "Normalized field coordinates"). Note that Zemax does not check to see if the specified Hx, Hy, Px, and Py coordinates are within the unit circle. For example a pupil coordinate of (1, 1) is actually outside the entrance pupil, but you will not get an error message when tracing these rays unless the rays cannot physically be traced.

Target, Value, and Weight

The target is the desired value of the specified parameter. The difference between the target and the value of the operand is squared, and summed over all operands to yield the merit function. The value of the target and the operand itself is unimportant in optimization, only the difference between the two. The larger the difference, the greater the contribution to the merit function.

The weight is the relative importance of that parameter. The weight can be any number, positive or negative. However, the optimizer will act somewhat differently if the weight is negative, zero, or positive.

Percent Contribution

This column lists the relative contribution of each operand to the overall merit function as a percentage. This column is not user editable.

Operand weights less than zero

When the weight is negative, the operand will be treated as a Lagrangian multiplier. The Lagrangian multipliers force the optimization algorithm to find a solution which exactly meets the specified constraint, regardless of the effect on the other operands. This is sometimes useful to exactly meet an optimization target, such as focal length or magnification. In some respects, this is similar to a weight of "infinity", however it is implemented in a way that is numerically more stable.

Because there is generally a non-linear relationship between the variables and the operand targets, Zemax may not converge to the exact target value in a single optimization cycle; however, multiple cycles will usually converge to the Lagrangian targets with extremely high precision in a few cycles if a solution exists. It is possible to define Lagrangian targets that cannot be met with the variables provided, especially if there is more than one Lagrangian target defined. For purposes of computing the overall merit function value, Zemax will use the absolute value of the weight.

For best results, use Lagrangian multipliers sparingly, if at all. Better optimization and adequate accuracy is usually just as easily achieved using heavier weights on those operands which require exact (or nearly so) values. If a Lagrangian multiplier operand does not exactly converge with a modest weight, such as -1, try a larger magnitude negative weight, such as -1000. This will allow the merit function to decrease while meeting the target exactly. Lagrangian multipliers should not be used with boundary operands, such CTGT, because appropriate derivatives cannot always be computed for these types of operands.

Operand weights equal to zero

When the weight is zero, the optimization algorithm calculates but ignores the operand. This is very useful for computing a result that does not have a specific target, but might be used elsewhere in the merit function; or if the value is used as a check or monitored parameter.

Operand weights greater than zero

If the weight is greater than zero, then the operand will be treated as an "aberration" to be minimized along with the merit function. The vast majority of operands should have positive weights.

Merit function definition

The merit function is defined as:

$$MF^2 = \frac{\sum W_i (V_i - T_i)^2}{\sum W_i},$$

where W is the absolute value of the weight of the operand, V is the current value, T is the target value, and the subscript i indicates the operand number (row number in the spreadsheet). The sum index "i" is normally over all operands in the merit function, however the merit function listing feature (see "Merit Function Listing") sums the user defined and default operands separately.

Optimization Operands

The following sections describe the available operands.
The first section is a "quick reference" table which categorizes the operands by general subject.
The second provides a detailed description of each operand, organized by category, and states which operands use which data fields. Names shown in **bold** define data values used to define the operand. These same names will appear in the editor column headings.
The third section provides the same detailed description of each operand, but lists the operands alphabetically in a single table, instead of by category as in the previous section.

If you are viewing this in an HTML dialog, expand the header in Contents tab to view the following subsections.

Optimization Operands Summary Table

This is a "quick reference" table which categorizes the operands by general subject.

Category	Related Operands
First-order optical properties	AMAG, ENPP, EFFL, EFLX, EFLY, EPDI, EXPD, EXPP, ISFN, ISNA, LINV, OBSN, PIMH, PMAG, POWF, POWP, POWR, SFNO, TFNO,

	WFNO
Aberrations	ABCD, ANAC, ANAR, ANAX, ANAY, ANCX, ANCY, ASTI, AXCL, BIOC, BIOD, BSER, COMA, DIMX, DISA, DISC, DISG, DIST, FCGS, FCGT, FCUR, LACL, LONA, OPDC, OPDM, OPDX, OSCD, PETC, PETZ, RSCE, RSCH, RSRE, RSRH, RWCE, RWCH, RWRE, RWRH, SMIA, SPCH, SPHA, TRAC, TRAD, TRAE, TRAI, TRAR, TRAX, TRAY, TRCX, TRCY, ZERN
MTF data	GMTA, GMTS, GMTT, MSWA, MSWS, MSWT, MTFA, MTFS, MTFT, MTHA, MTHS, MTHT
PSF/Strehl Ratio data	STRH
Encircled energy	DENC, DENF, ERFP, GENC, GENF, XENC, XENF
Constraints on lens data	COGT, COLT, COVA, CTGT, CTLT, CTVA, CVGT, CVLT, CVVA, BLTH, DMGT, DMLT, DMVA, ETGT, ETLT, ETVA, FTGT, FTLT, MNCA, MNCG, MNCT, MNCV, MNEA, MNEG, MNET, MNPD, MXCA, MXCG, MXCT, MXCV, MXEA, MXEG, MXET, MNSD, MXSD, TGTH, TTGT, TTHI, TTLT, TTVA, XNEA, XNET, XNEG, XXEA, XXEG, XXET, ZTHI
Constraints on lens properties	CVOL, MNDT, MXDT, SAGX, SAGY, SSAG, STHI, TMAS, TOTR, VOLU, NORX, NORY, NORZ, NORD, SCUR, SDRV
Constraints on parameter data	PMGT, PMLT, PMVA.
Constraints on extra data	XDGT, XDLT, XDVA
Constraints on glass data	GCOS, GTCE, INDX, MNAB, MNIN, MNPD, MXAB, MXIN, MXPD, RGLA
Constraints on paraxial ray data	PANA, PANB, PANC, PARA, PARB, PARC, PARR, PARX, PARY, PARZ, PATX, PATY, YNIP
Constraints on real ray data	CEHX, CEHY, CENX, CENY, CNAX, CNAY, CNPX, CNPY, DXDX, DXDY, DYDX, DYDY, HHCN, IMAE, MNRE, MNRI, MXRE, MXRI, OPTH, PLEN, RAED, RAEN, RAGA, RAGB, RAGC, RAGX, RAGY, RAGZ, RAID, RAIN, RANG, REAA, REAB, REAC, REAR, REAX, REAY, REAZ, RENA, RENB, RENC, RETX, RETY
Constraints on element positions	GLCA, GLCB, GLCC, GLCR, GLCX, GLCY, GLCZ
Changing system data	CONF, IMSF, PRIM, SVIG, WLEN, CVIG, FDMO, FDRE
General math operands	ABSO, ACOS, ASIN, ATAN, CONS, COSI, DIFF, DIVB, DIVI, EQUA, LOGE, LOGT, MAXX, MINN, OPGT, OPLT, OPVA, OSUM, PROB,

	PROD, QSUM, RECI, SQRT, SUMM, SINE, TANG, ABGT, ABLT
Multi-configuration (zoom) data	CONF, MCOL, MCOG, MCOV, ZTHI
Gaussian beam data	GBPD, GBPP, GBPR, GBPS, GBPW, GBPZ, GBSD, GBSP, GBSR, GBSS, GBSW
Gradient index control operands	DLTN, GRMN, GRMX, InGT, InLT, InVA, LPTD
Foucault analysis	FOUC
Ghost focus control	GPIM, GPRT, GPRX, GPRY, GPSX, GPSY
Fiber coupling operands	FICL, FICP, POPD
Relative illumination operand	RELI, EFNO
Optimization with ZPL macros	ZPLM
User defined operands	UDOP
Merit function control operands	BLNK, DMFS, ENDX, GOTO, OOFF, SKIN, SKIS, USYM
Constraints on non-sequential object data.	FREZ, NPGT, NPLT, NPVA, NPXG, NPXL, NPXV, NPYG, NPYL, NPYV, NPZG, NPZL, NPZV, NSRM, NTXG, NTXL, NTXV, NTYG, NTYL, NTYV, NTZG, NTZL, NTZV
Non-sequential ray tracing and detector operands.	NSDC, NSDD, NSDE, NSDP, NSLT, NSRA, NSRM, NSRW, NSST, NSTR, NSTW, REVR
Constraints on construction optics for optically fabricated holograms	CMFV
Constraints on optical coatings, polarization ray trace data	CMGT, CMLT, CMVA, CODA, CEGT, CELT, CEVA, CIGT, CILT, CIVA
Physical Optics Propagation (POP) results	POPD, POPI
Best Fit Sphere data	BFSD
Tolerance sensitivity data	TOLR
Thermal Coefficient of Expansion data	TCGT, TCLT, TCVA

Optimization Operands by Category

This section provides a detailed description of each operand, organized by category, and states which operands use which data fields. Each category is a different heading, and is followed by a table with the corresponding operands. Names shown in **bold** define data values used to define the operand. These same names will appear in the editor column headings.

First-order optical properties

Operands for first-order optical properties	
AMAG, ENPP, EFFL, EFLX, EFLY, EPDI, EXPD, EXPP, ISFN, ISNA LINV, OBSN, PIMH, PMAG, POWF, POWP, POWR, SFNO, TFNO, WFNO	
NAME	**Description**
AMAG	Angular magnification. This is the ratio of the image to object space paraxial chief ray angles at the wavelength defined by **Wave**. Not valid for non-paraxial systems.
ENPP	Entrance pupil position in lens units, with respect to the first surface. This is the paraxial pupil position, valid only for centered systems.
EFFL	Effective focal length in lens units. The wavelength used is defined by **Wave**. This is paraxial, and may not be accurate for non-paraxial systems.
EFLX	Effective focal length in the local x plane of the range of surfaces defined by **Surf1** and **Surf2** at the primary wavelength.
EFLY	Effective focal length in the local y plane of the range of surfaces defined by **Surf1** and **Surf2** at the primary wavelength.
EPDI	Entrance pupil diameter in lens units.
EXPD	Exit pupil diameter in lens units. This is the paraxial pupil diameter, valid only for centered systems.
EXPP	Exit pupil position in lens units, with respect to the image surface. This is the paraxial pupil position, valid only for centered systems.

ISFN	Image space F/#. This operand is the paraxial infinite conjugate F/#. See WFNO.
ISNA	Image space numerical aperture. This operand is the paraxial image space na at the defined conjugates. See ISFN.
LINV	Lagrange (or optical) invariant of system in lens units at the wavelength defined by **Wave**. The paraxial marginal and chief ray data are used to compute this value.
OBSN	Object space numerical aperture. This is only useful for finite conjugate systems, and is calculated on axis at the primary wavelength.
PIMH	Paraxial image height at the paraxial image surface at the wavelength defined by **Wave**. Not valid for non-paraxial systems.
PMAG	Paraxial magnification. This is the ratio of the paraxial chief ray height on the paraxial image surface to the object height at the wavelength defined by **Wave**. Only useful for finite conjugate systems. Note the paraxial image surface is used even if the system is not at paraxial focus.
POWF	Power at a field point. Computes the power or effective focal length (EFL) after refraction from the surface defined by **Surf** at the wavelength defined by **Wave** for any point in the field. For the **Data** parameter, use 0 for spherical, 1 for cylinder, 2 for max, 3 for min, 4 for tangential, 5 for sagittal, 6 for y, 7 for x, and 8 for astigmatic power in diopters. To get the EFL values, add 9 to these codes. For example, for tangential EFL use **Data** = 13. For cylinder, the EFL value is the difference between the maximum and minimum focal lengths. For astigmatic, the EFL is the difference between the x and y direction focal lengths (x-y). For a full description of this type of analysis, see "Power Field Map". See "Hx, Hy, Px, and Py".
POWP	Power at a point in the pupil. Computes the power or effective focal length (EFL) after refraction from the surface defined by **Surf** at the wavelength defined by **Wave** for any point in the field, at any point in the pupil. For the **Data** parameter, use 0 for spherical, 1 for cylinder, 2 for max, 3 for min, 4 for tangential, 5 for sagittal, 6 for y, 7 for x, and 8 for astigmatic power in diopters. To get the EFL values, add 9 to these codes. For example, for tangential EFL use **Data** = 13. For cylinder, the EFL value is the difference between the maximum and minimum focal lengths. For astigmatic, the EFL is the difference between the x and y direction focal lengths (x-y). For a full description of this type of analysis, see "Power Pupil Map". See "Hx, Hy, Px, and Py".
POWR	The surface power (in inverse lens units) of the surface defined by **Surf** at the wavelength defined by **Wave**. This operand only works for standard surfaces.

SFNO	Sagittal working F/#, computed at the field point defined by **Field** and the wavelength defined by **Wave**. See TFNO.
TFNO	Tangential working F/#, computed at the field point defined by **Field** and the wavelength defined by **Wave**. See SFNO.
WFNO	Working F/#. See "Working F/#", and ISFN, SFNO, and TFNO.

Aberrations

Operands for aberrations

ABCD, ANAC, ANAR, ANAX, ANAY, ANCX, ANCY, ASTI, AXCL, BIOC, BIOD, BSER, COMA, DIMX, DISA, DISC, DISG, DIST, FCGS, FCGT, FCUR, LACL, LONA, OPDC, OPDM, OPDX, OSCD, PETC, PETZ, RSCE, RSCH, RSRE, RSRH, RWCE, RWCH, RWRE, RWRH, SMIA, SPCH, SPHA, TRAC, TRAD, TRAE, TRAI, TRAR, TRAX, TRAY, TRCX, TRCY, ZERN

NAME	Description
ABCD	The ABCD values used by the grid distortion feature to compute generalized distortion. See "Grid Distortion". The reference field number is defined by **Ref Fld**. The wavelength number is defined by **Wave**. **Data** is 0 for A, 1 for B, 2 for C, and 3 for D. See also "DISA".
ANAC	Angular aberration radial direction measured in image space with respect to the centroid at the wavelength defined by **Wave**. This quantity is defined as: $$\varepsilon = \sqrt{(l - l_c)^2 + (m - m_c)^2}$$ where l and m are the x and y direction cosines of the ray and the c subscript indicates the centroid. See "Hx, Hy, Px, and Py".
ANAR	Angular aberration radius measured in image space at the wavelength defined by **Wave** with respect to the primary wavelength chief ray. This quantity is defined as: $$\varepsilon = \sqrt{(l - l_c)^2 + (m - m_c)^2}$$ where l and m are the x and y direction cosines of the ray and the c subscript indicates the chief ray. See "Hx, Hy, Px, and Py" .

| ANAX | Angular aberration x direction measured in image space at the wavelength defined by **Wave** with respect to the primary wavelength chief ray. This quantity is defined as:

$$\varepsilon = l - l_c$$

where l is the x direction cosine of the ray and the c subscript indicates the chief ray. See "Hx, Hy, Px, and Py". |
|---|---|
| ANAY | Angular aberration y direction measured in image space at the wavelength defined by **Wave** with respect to the primary wavelength chief ray. This quantity is defined as:

$$\varepsilon = m - m_c$$

where m is the y direction cosine of the ray and the c subscript indicates the chief ray. See "Hx, Hy, Px, and Py". |
| ANCX | Angular aberration x direction measured in image space at the wavelength defined by **Wave** with respect to the centroid. This quantity is defined as:

$$\varepsilon = l - l_c$$

where l is the x direction cosine of the ray and the c subscript indicates the centroid. ANCX has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py". |
| ANCY | Angular aberration y direction measured in image space at the wavelength defined by **Wave** with respect to the centroid. This quantity is defined as:

$$\varepsilon = m - m_c$$

where m is the y direction cosine of the ray and the c subscript indicates the centroid. ANCY has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py". |
| ASTI | Astigmatism in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order astigmatism calculated from the Seidel coefficients, and is not valid for non-paraxial systems. |
| AXCL | Axial color, measured in lens units for focal systems and diopters for afocal systems. This is the image separation between the two wavelengths defined by **Wave1** and **Wave2**. If **Zone** is zero, paraxial rays are used to determine the paraxial image locations. If **Zone** is greater than 0.0 and less than or equal to 1.0, real marginal rays are used to determine the image locations. In this case, **Zone** corresponds to the Py coordinate of the real marginal ray. |

	See "Hx, Hy, Px, and Py".
BIOC	Biocular Convergence. Returns the convergence between two eye configurations in milliradians. The left and right eye configurations are specified using the **Left** and **Right** values. The other parameters are: **Wave**: The wavelength number to use. **UseCos**: If 0 field units are degrees, otherwise field is in direction cosine units. **Xang**/**Yang**: The X and Y angle or cosines at which to compute the convergence. If the chief rays from both configurations at the specified angles do not pass through to the image without vignetting, an error is reported. See "Divergence/Convergence" for more information and important assumptions.
BIOD	Biocular Dipvergence. Returns the dipvergence between two eye configurations in milliradians. See BIOC above for details.
BSER	Boresight error. Boresight error is defined as the radial chief ray coordinate traced for the on axis field and wavelength defined by **Wave** divided by the effective focal length. This definition yields a measure of the angular deviation of the image.
COMA	Coma in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order coma calculated from the Seidel coefficients, and is not valid for non-paraxial systems.
DIMX	Distortion maximum. This is similar to DIST, except it specifies only an upper bound for the absolute value of the distortion. **Field** can be zero, which specifies that the maximum field coordinate be used, or any valid field number. Note the maximum distortion does not always occur at the maximum field coordinate. The distortion is calculated at the wavelength defined by **Wave**. If **Absolute** is 0, the value returned is in units of percentage. If **Absolute** is 1, the distortion is given as an absolute length rather than a percentage. This operand may not be valid for non-rotationally symmetric systems.
DISA	Distortion, ABCD. This operand computes the Radial, X, or Y direction distortion relative to the reference field (**Ref Fld**), for the chief ray at the wavelength defined by **Wave. Data** is 0 for radial distortion, 1 for X direction distortion, and 2 for Y direction distortion. The distortion is computed for the chief ray at the field point defined by **Field**. The **A**, **B**, **C**, **D** values are user defined. The distortion is computed in the same manner as the grid distortion feature (see "Grid Distortion"). The key difference between this operand and DISG is that the ABCD values are user defined. See also "ABCD" and "DISG".

DISC	Distortion, calibrated. This operand computes the calibrated f-theta distortion across the y-field of view at the wavelength defined by **Wave**, and returns the absolute value of the maximum deviation from linearity of the f-theta condition.
	If **Absolute** is 0, the value returned is in units of percentage. If **Absolute** is 1, the distortion is given as an absolute length (for focal systems) or an absolute deviation in cosine space (for afocal systems) rather than a percentage. This operand is useful for designing f-theta lenses.
DISG	Generalized distortion, either in percent or as an absolute distance. This operand computes the distortion for any ray in the pupil, from anywhere in the field, at the wavelength defined by **Wave**, using the field point defined by **Field** as a reference. The method used and assumptions made are the same as for the grid distortion plot, see "Grid Distortion". DISG cannot be calculated if the field units are angles and the maximum angle equals or exceeds 90 degrees. DISG assumes the predicted magnification is not symmetric.
	If the field is defined in terms of angles, the normalized field coordinates **Hx** and **Hy** are defined differently than for other Zemax features. In this case only, **Hx** and **Hy** correspond to divisions on a uniform projected grid in object space, centered on the reference field coordinate. These normalized field coordinates are defined as
	$$H = \frac{\tan(\theta)}{\tan(\theta_M)}$$
	where θ is the angle relative to the reference field chief ray and θ_M is the maximum field angle (see "Maximum field"). These definitions are needed to support the reference field feature.
	If **Wave** is a positive number, DISG returns the distortion as a percentage. If **Wave** is a negative number, the absolute value of **Wave** is used to define the wavelength and the returned value is in units of absolute length rather than percentage.
	As with all distortion concepts, the best way to avoid confusion and misleading results is to use finite object distances and object heights to define fields rather than field angles.
	See "Hx, Hy, Px, and Py".
DIST	Distortion in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. This is the third order distortion calculated from the Seidel coefficients (see "Seidel Coefficients"), and is not valid for non-paraxial systems.
	If **Surf** is zero, the distortion is given in percent instead (see "Field Curvature/Distortion" for a detailed definition).
	If **Absolute** is set to 1, and the surface number is zero, the distortion is given as an absolute length rather than a percentage.
	See also DISG.

FCGS	Generalized field curvature, sagittal. The field curvature value for any field point, at the wavelength defined by **Wave**. The value is generalized to return reasonable results even for non-rotationally symmetric systems; see "Field Curvature/Distortion". See "Hx, Hy, Px, and Py".
FCGT	Generalized field curvature, tangential; see FCGS.
FCUR	Field curvature in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order field curvature calculated from the Seidel coefficients, and is not valid for non-paraxial systems.
LACL	Lateral color. For focal systems, this is the y-distance between the paraxial chief ray intercepts of the two extreme wavelengths defined by **Minw** and **Maxw**, measured in lens units. For afocal systems, this is the angle in afocal mode units between the paraxial chief rays of the two extreme wavelengths defined by **Minw** and **Maxw**.
LONA	Longitudinal aberration, measured in lens units for focal systems and diopters for afocal systems. This is the defocus from the current image to the image at the wavelength defined by **Wave** and pupil zone defined by **Zone**. If **Zone** is zero, paraxial rays are used to determine the paraxial image locations. If **Zone** is greater than 0.0 and less than or equal to 1.0, real marginal rays are used to determine the image locations. In this case, **Zone** corresponds to the Py coordinate of the real marginal ray. See AXCL.
OPDC	Optical path difference with respect to chief ray in waves at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py" .
OPDM	Optical path difference with respect to the mean OPD over the pupil at the wavelength defined by **Wave**. OPDM has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py".
OPDX	Optical path difference with respect to the mean OPD over the pupil with tilt removed at the wavelength defined by **Wave**. OPDX has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py".
OSCD	Offense against the sine condition (OSC) at the wavelength defined by **Wave**. There are two definitions for OSC supported. The first definition is as described in Welford, Aberrations of Optical Systems (see "REFERENCES ON LENS DESIGN"). This definition is used if **Zone** is zero. An alternate definition due to Prof. Roland Shack

	which supports computation of OSC as a function of pupil zone and uses only real rays is available. This definition is used if **Zone** is not zero. In this case, **Zone** corresponds to the Py coordinate of the real marginal ray. The two methods will give very similar results for systems with modest F/#'s and aberrations when **Zone** is 1.0 for the alternate definition. This operand has no meaning if the system is not axially symmetric.
PETC	Petzval curvature in inverse lens units at the wavelength defined by **Wave**. Not valid for non-paraxial systems.
PETZ	Petzval radius of curvature in lens units at the wavelength defined by **Wave**. Not valid for non- paraxial systems.
RSCE	RMS spot radius with respect to the centroid in lens units. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RSCH	RMS spot radius with respect to the chief ray in lens units. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RSRE	RMS spot radius with respect to the centroid in lens units. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RSRH	RMS spot radius with respect to the chief ray in lens units. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RWCE	RMS wavefront error with respect to the centroid in waves. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py", and "OPTIMIZATION REFERENCE POINTS".
RWCH	RMS wavefront error with respect to the chief ray in waves. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular

	pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py" , and "OPTIMIZATION REFERENCE POINTS".
RWRE	RMS wavefront error with respect to the centroid in waves. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py", and "OPTIMIZATION REFERENCE POINTS".
RWRH	RMS wavefront error with respect to the chief ray in waves. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py", and "OPTIMIZATION REFERENCE POINTS".
SMIA	SMIA-TV Distortion. **Field** is the zero distortion reference field position, or use zero to indicate the field position (0, 0). **Wave** is the wavelength number, or use zero for the primary wavelength. **X- Width** and **Y-Width** are the full field width in field units. For more information see "SMIA-TV Distortion".
SPCH	Spherochromatism in lens units. This is the difference between the real marginal axial color and the paraxial axial color of the two extreme wavelengths defined by **Minw** and **Maxw**. The distance is measured along the Z axis. **Zone** defines the zone for which the real marginal axial color is computed. **Zone** corresponds to the Py coordinate of the real marginal ray. Not valid for non-paraxial systems.
SPHA	Spherical aberration in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order spherical aberration calculated from the Seidel coefficients, and is not valid for non-paraxial systems.
TRAC	Transverse aberration radial direction measured in image space with respect to the centroid for the wavelength defined by **Wave**. Unlike most other operands, TRAC critically depends upon the placement of other TRAC operands within the Merit Function Editor to work correctly. TRAC operands must be grouped together by field position and wavelength. Zemax traces all TRAC rays with a common field point together, and then uses the collective data to compute the centroid of all the rays. Each ray individually is then referenced to the computed centroid. This operand should only be entered into the Merit Function Editor by the Sequential Merit Function tool, and is

	not recommended for use directly by the user. See "Hx, Hy, Px, and Py".
TRAD	The x component of the TRAR only. TRAD has the same restrictions that TRAC does; see TRAC for a detailed discussion.
TRAE	The y component of the TRAR only. TRAE has the same restrictions that TRAC does; see TRAC for a detailed discussion.
TRAI	Transverse aberration radius measured at the surface defined by **Surf** at the wavelength defined by **Wave** with respect to the chief ray. Similar to TRAR, except a surface other than the image surface may be specified. See "Hx, Hy, Px, and Py".
TRAR	Transverse aberration radial direction measured in image space at the wavelength defined by **Wave** with respect to the chief ray. See ANAR. See "Hx, Hy, Px, and Py".
TRAX	Transverse aberration x direction measured in image space at the wavelength defined by **Wave** with respect to the chief ray. See "Hx, Hy, Px, and Py".
TRAY	Transverse aberration y direction measured in image space at the wavelength defined by **Wave** with respect to the chief ray. See "Hx, Hy, Px, and Py".
TRCX	Transverse aberration x direction measured in image space with respect to the centroid. TRCX has the same restrictions that TRAC does; see TRAC for a detailed discussion.
TRCY	Transverse aberration y direction measured in image space with respect to the centroid. TRCY has the same restrictions that TRAC does; see TRAC for a detailed discussion.
ZERN	Zernike Fringe coefficient. The parameters are: **Term**: The Zernike term number (1 - 37 for fringe, 1 - 231 for standard or annular). **Wave**: The wavelength number. **Samp**: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. **Field**: The field number. **Type**: The Zernike type (0 for fringe, 1 for standard, 2 for annular). **Epsilon**: The

obscuration ratio (for annular coefficients only). **Vertex?**: If 1, the OPD is referenced to the surface vertex.

The **Term** value, if negative or zero, may also be used to return other data from the Zernike fitting as follows:

-8: Peak to Valley OPD (to centroid)

-7: Peak to Valley OPD (to chief)

-6: RMS to zero reference (unused by Zemax)

-5: RMS to chief ray

-4: RMS to centroid

-3: Variance

-2: Strehl Ratio

-1: RMS fit error

0: Maximum single point fit error

Note that if you use multiple ZERN operands which only differ in the **Term** value, they should be placed on adjacent lines in the editor so Zemax only does the fitting once; otherwise, the computation is slower.

Multiple Zernike terms are always used in the fitting process, even if only one coefficient is requested. The maximum number of terms used in the computation depends on the **Type** and the **Term** settings. The minimum number of terms used for all types is 11. This means that **Term** is only used if set greater than or equal to 11. If **Type** is standard or annular, the maximum term number computed is set equal to the largest term requested by any **Term** value in adjacent ZERN operands.

MTF data

Operands for MTF data

GMTA, GMTS, GMTT, MSWA, MSWS, MSWT, MTFA, MTFS, MTFT, MTHA, MTHS, MTHT

NAME	Description
GMTA	Geometric MTF average of sagittal and tangential response. The parameters are: **Samp**: Higher sampling yields higher accuracy at the expense of computation time. To confirm the computation has acceptable accuracy, start at 1, and increment the

	sampling until the results change by less than the desired accuracy. Note that extreme precision is not required for good optimization results; three significant figures is usually adequate.
	Wave: The wavelength number to use (use 0 for polychromatic).
	Field: The field number.
	Freq: The spatial frequency in MTF units (see "MTF Units").
	!Scl: If zero, then the diffraction limit will be used to scale the results (recommended) otherwise, no scaling is done.
	If **Grid** is zero, a fast, sparse sampling integration method is used to compute the MTF. The fast Geometric MTF algorithm is only accurate for systems with circular or elliptical pupils with modest or no apodization. For systems that violate this assumption, set **Grid** equal to 1. The fast sampling method used by GMTA, GMTS, and GMTT is not directly related to the Geometric MTF Analysis feature. Because only a single spatial frequency is required, the method of computation used by the MTF operands is different, and generally much faster, than the algorithm used by the analysis feature. To select the alternate grid-based algorithm used by the Geometric MTF analysis feature, set **Grid** equal to 1. The grid-based algorithm is usually slower than the default algorithm if the MTF is reasonably good (greater than 5%), but the grid algorithm converges faster if the aberrations are very large and the resulting MTF is very low.
	If both the tangential and sagittal MTF are needed; place the GMTT and GMTS operands on adjacent lines and they will be computed simultaneously. The Geometric MTF, though approximate, can usually be computed more quickly than the Diffraction MTF, and is therefore useful for optimization. See "Performing an optimization".
GMTS	Geometric MTF sagittal response. See GMTA.
GMTT	Geometric MTF tangential response. See GMTA.
MSWA	Modulation square-wave transfer function, average of sagittal and tangential. See MTFA for details.
MSWS	Modulation square-wave transfer function, sagittal. See MTFA for details.
MSWT	Modulation square-wave transfer function, tangential. See MTFA for details.
MTFA	Diffraction modulation transfer function, average of sagittal and tangential. The parameters are: **Samp**: Higher sampling yields higher accuracy at the expense of computation time. To confirm the computation has acceptable accuracy, start at 1, and increment the sampling until the results change by less than the desired accuracy. Note that extreme precision is not required for good optimization results; three significant figures is usually adequate. There are two algorithms available for computing the MTF. If **Grid** is zero (recommended), a fast, sparse sampling integration method is used to compute the MTF. The fast sampling method used by MTFA, MTFS,

	and MTFT is not directly related to the MTF Analysis feature. Because only a single spatial frequency is required, the method of computation used by the MTF operands is different, and generally much faster, than the algorithm used by the analysis feature. To select the alternate grid-based algorithm used by the MTF analysis feature, set **Grid** equal to 1. The grid-based algorithm is usually slower than the default algorithm if the MTF is reasonably good (greater than 5%), but the grid algorithm converges faster if the aberrations are very large and the resulting MTF is very low. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Freq**: The spatial frequency in MTF units (see "MTF Units"). If the sampling is set too low for accurate computation of the MTF, then the MTF operands all return zero. If both the tangential and sagittal MTF are needed; place the MTFT and MTFS operands on adjacent lines and they will be computed simultaneously. See "Performing an optimization". **Data Type**: Specifies the data to be returned. An input of 0 will return the modulation amplitude; an input of 1 will return the real part; an input of 2 will return the imaginary part; an input of 3 will return the phase. This input is only available for the MTFA, MTFS, and MTFT operands (and not the equivalent square-wave operands).
MTFS	Modulation transfer function, sagittal. See MTFA for details.
MTFT	Modulation transfer function, tangential. See MTFA for details.
MTHA	Huygens Modulation transfer function, average of sagittal and tangential. This operand computes the diffraction MTF using the Huygens method (see "Huygens MTF"). The parameters are: **Samp**: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. The sampling is assumed to be the same for both pupil and image. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Freq**: The spatial frequency in MTF units (see "MTF Units"). If the sampling is set too low for accurate computation of the MTF, then the MTF operands all return zero. **Pol?:** Set to 0 to ignore polarization and 1 to consider it. **All Conf?**: Set to 0 to use the current configuration (defined by the last CONF operand preceding this operand), and 1 to sum over all configurations. See "Huygens MTF" for a full discussion of this option. **Ima Delta**: The image delta in micrometers used for the computation. If zero, the default

	image delta is used.
	If both the tangential and sagittal MTF are needed; place the MTHT and MTHS operands on adjacent lines and they will be computed simultaneously. See "Performing an optimization".
MTHS	Huygens Modulation transfer function, sagittal. See MTHA for details.
MTHT	Huygens Modulation transfer function, tangential. See MTHA for details.

PSF/Strehl Ratio data

Operands for PSF/Strehl Ratio data

STRH

NAME	Description
STRH	Strehl Ratio. This operand computes the Strehl Ratio using the Huygens PSF computation (see "Huygens PSF"). The parameters are:
	Samp: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. The sampling is assumed to be the same for both pupil and image.
	Wave: The wavelength number to use (use 0 for polychromatic).
	Field: The field number.
	Pol?: Set to 0 to ignore polarization and 1 to consider it.
	All Conf?: Set to 0 to use the current configuration (defined by the last CONF operand preceding this operand), and 1 to sum over all configurations. See "Huygens PSF" for a full discussion of this option.
	By default, the image delta used in the Huygens PSF computation for the evaluation of this operand is 2x smaller than the nominal default image delta. See "Huygens PSF" on for the formula used to calculate the nominal default image delta. This is done to zoom in on the peak of the PSF, allowing for a more accurate computation of the Strehl ratio. As a result, the value reported will, in general, be different than the value reported using the nominal default image delta in the Huygens PSF analysis.

Encircled energy

Operands for encircled energy

DENC, DENF, ERFP, GENC, GENF, XENC, XENF

NAME	Description
DENC	Diffraction Encircled Energy (distance). This operand computes the distance to the fraction of diffraction encircled, ensquared, x only, or y only (enslitted) energy defined by **Frac**. For focal mode, the units are micrometers. For afocal mode, the units are afocal mode units. The other parameters are: **Samp:** The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Type**: 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. **Refp**: The reference point/algorithm to use. For FFT encircled energy, use **Refp** = 0 for chief ray, 1 for centroid, and 2 for vertex. For Huygens encircled energy, use **Refp** = 3 for chief ray, 4 for centroid, and 5 for vertex. When using the Huygens method, **I Samp** refers to the Huygens image sampling (1 for 32 x 32, 2 for 64 x 64, etc.) and **I Delta** refers to the Huygens image delta. For a detailed description of the Huygens image sampling and image delta, see "Encircled Energy". If the sampling is too low, the radius returned is 1e+10. See also DENF, GENC and XENC.
DENF	Diffraction Encircled Energy (fraction). This operand computes the fraction of diffraction encircled, ensquared, x only, or y only (enslitted) energy at a given distance from the reference point defined by **Dist**. For focal mode, the distance units are micrometers. For afocal mode, the distance units are afocal mode units. The options and settings are identical to DENC, except **Dist**, which here is used as the distance at which the fraction of energy is desired. See also DENC, GENC, GENF, and XENC. If the **Dist** value defined is beyond the point where the encircled energy is very close to 100%, the fraction returned is 1e+10; this is done for optimization efficiency.
ERFP	Edge Response Function Position. This operand computes the x or y position of the point at which the edge response function reaches a certain relative value. For details on the edge response function calculation, see "Geometric Line/Edge Spread". The **Sampling** value is 1 for 32x32, 2 for 64x64, etc. **Wave** is the wavelength number or 0 for polychromatic. **Field** is the field number. **Type** determines that data to be returned. If **Type** is 0 or 1, the x position (for edges parallel to the local y axis) or y position (for edges parallel to the local x axis) relative to the chief ray in lens units is returned. If **Wave** is zero, the primary wavelength chief ray is the reference point. If **Type** is 2 or 3, the x or y position in lens units relative to the surface vertex is returned. **Fraction** is the relative value of the edge response curve, and must be between 0.01 and 0.99. **Max Radius** is the maximum radial size of the integration window in micrometers. If **Max Radius** is zero a default value is used, and this is the recommended setting in most cases.

	Note that the edge response function is normally defined with the "bright" side of the edge being on the + side of the integration coordinate, which means the edge response goes to 1 as the coordinate becomes more positive. To compute results for the reversed edge orientation, with the bright side on the negative side of the coordinate, use the value (1-fraction) instead of (fraction). For example, to get the 80% response coordinate for a reversed edge, use **Fraction** = 0.20. If afocal mode is used, all returned values are in afocal analysis units.
GENC	Geometric Encircled Energy (distance). This operand computes the distance to the fraction of geometric encircled, ensquared, x only, or y only (enslitted) energy defined by **Frac**. For focal mode, the units are micrometers. For afocal mode, the units are afocal mode units. The other parameters are: **Samp**: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Type**: 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. **Refp**: The reference point to use. Use 0 for chief ray, 1 for centroid, 2 for vertex, and 3 for middle of the spot. **No Diff Lim**: If 0, the results are scaled by the diffraction limit, otherwise, no accounting of diffraction is done. See also GENF, DENC, DENF, and XENC.
GENF	Geometric Encircled Energy (fraction). This operand computes the fraction of geometric encircled, ensquared, x only, or y only (enslitted) energy at a given distance from the reference point defined by **Dist**. The options and settings are identical to GENC, except **Dist**, which here is used as the distance at which the fraction of energy is desired. See also GENC, DENC, DENF, and XENC.
XENC	Extended source encircled energy (distance). This operand computes the distance in micrometers to the specified fraction of extended source geometric encircled energy, using whatever the current default settings are. To use this operand, first define the settings on the extended source encircled energy feature as desired, then press Save on the settings box. The only settings that are overwritten are those set by **Type** and **Wave**. **Type** is 1 for encircled, 2 for x only, 3 for y only, 4 for ensquared, 5 for x distribution, and 6 for y distribution. **Wave** is the input wavenumber. **Frac** is the fraction of energy desired, and must be between zero and 1, exclusive. **Frac** is ignored for type 5 or 6; for these types the returned valued is the full width half max independent of **Frac**.

	Max Radius is the maximum distance in radial distance in micrometers. If this value is zero, the default setting is used. See also XENF, DENC, DENF, GENC, and GENF.
XENF	Extended source encircled energy (fraction). This operand computes the fraction of extended source geometric encircled, ensquared, x only, or y only (enslitted) energy at a given distance from the reference point. The **Type** is 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. **Wave** is the input wavenumber. The options and settings are identical to XENC, except **Dist**, which here is used as the distance at which the fraction of energy is desired. See also XENC, GENC, GENF, DENC, and DENF.

Constraints on lens data

Operands for constraints on lens data

COGT, COLT, COVA, CTGT, CTLT, CTVA, CVGT, CVLT, CVVA, BLTH, DMGT, DMLT, DMVA, ETGT, ETLT, ETVA, FTGT, FTLT, MNCA, MNCG, MNCT, MNCV, MNEA, MNEG, MNET, MNPD, MXCA, MXCG, MXCT,MXCV, MXEA, MXEG, MXET, MNSD, MXSD, TGTH, TTGT, TTHI, TTLT, TTVA, XNEA, XNET, XNEG, XXEA, XXEG, XXET, ZTHI

NAME	Description
COGT	Boundary operand that constrains the conic of the surface defined by **Surf** to be greater than the specified target value.
COLT	Boundary operand that constrains the conic of the surface defined by **Surf** to be less than the specified target value.
COVA	Conic value. Returns the conic constant of the surface defined by **Surf**.
CTGT	Center thickness greater than. This boundary operand constrains the center thickness of the surface defined by **Surf** to be greater than the specified target value. See also MNCT.

CTLT	Center thickness less than. This boundary operand constrains the center thickness of the surface defined by **Surf** to be less than the specified target value. See also MXCT.
CTVA	Center thickness value. Constrains the center thickness of the surface defined by **Surf** to be equal to the specified target value.
CVGT	Curvature greater than. This boundary operand constrains the curvature of the surface defined by **Surf** to be greater than the target value.
CVLT	Curvature less than. This boundary operand constrains the curvature of the surface defined by **Surf** to be less than the target value.
CVVA	Curvature value. This operand constrains the curvature of the surface defined by **Surf** to be equal to the specified target value.
BLTH	Blank thickness. Computes the minimum thickness of the glass blank required to create the volume following the surface defined by **Surf**. The sag of the surface and the one following are computed at 200 radial points along axes defined by the **Code** value. Use **Code** = 0 for +Y axis, 1 for +X axis, 2 for -Y axis, 3 for -X axis, and 4 for all four axes.
DMGT	Diameter greater than. This boundary operand constrains the diameter of the surface defined by **Surf** to be greater than the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.
DMLT	Diameter less than. This boundary operand constrains the diameter of the surface defined by **Surf** to be less than the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.
DMVA	Diameter value. This operand constrains the diameter of the surface defined by **Surf** to be equal to the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.
ETGT	Edge thickness greater than. This boundary operand constrains the edge thickness of the surface defined by **Surf** to be greater than the specified target value. The edge thickness is calculated along the +y axis if **Code** is zero, the +x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. See also MNET.
ETLT	Edge thickness less than. This boundary operand constrains the edge thickness of the surface defined by **Surf** to be less than the specified target value. The edge thickness is calculated along the +y axis if **Code** is zero, the +x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. See also MXET.

ETVA	Edge thickness value. Constrains the edge thickness of the surface defined by **Surf** to be equal to the specified target value. The edge thickness is calculated along the +y axis if **Code** is zero, the
	+x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. See also MNET.
FTGT	Full thickness greater than. This boundary operand constrains the full thickness of surface **Surf** to be greater than the specified target value. The full thickness is computed at 200 points between the vertex and edge along the +y radial direction, including the sag of the surface and the sag of the next surface. The operand is useful for constraining surfaces which do not have their minimum or maximum thickness at the center or edge, but at some intermediate zone. See FTLT.
FTLT	Full thickness less than. See FTGT.
MNCA	Minimum center thickness air. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be greater than the specified target value. See also MNCT and MNCG. This operand controls multiple surfaces simultaneously.
MNCG	Minimum center thickness glass. This boundary operand constrains each of the thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be greater than the target value. See also MNCT and MNCA. This operand controls multiple surfaces simultaneously.
MNCT	Minimum center thickness. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MNCG and MNCA. This operand controls multiple surfaces simultaneously.
MNCV	Minimum curvature. This boundary operand constrains each of the curvatures of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MXCV. This operand controls multiple surfaces simultaneously.
MNEA	Minimum edge thickness air. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be greater than the specified target value. See also MNET, MNEG, ETGT, and XNEA. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XNEA for constraining non-rotationally symmetric surfaces.
	Zone, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.

MNEG	Minimum edge thickness glass. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be greater than the specified target value. See also MNET, MNEA, ETGT, and XNEG. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XNEG for constraining non-rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MNET	Minimum edge thickness. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MNEG, MNEA, ETGT, and XNET. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XNET for constraining non-rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MNPD	Minimum $\Delta P_{g,F}$. This boundary operand constrains the deviation of the partial dispersion of surfaces between **Surf1** and **Surf2** to be greater than the specified target value. See also MXPD. This operand controls multiple surfaces simultaneously.
MXCA	Maximum center thickness air. This boundary operand constrains each of the thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be less than the target value. See also MXCT and MXCG. This operand controls multiple surfaces simultaneously.
MXCG	Maximum center thickness glass. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be less than the target value. See also MXCT and MXCA. This operand controls multiple surfaces simultaneously.
MXCT	Maximum center thickness. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also MXCG and MXCA. This operand controls multiple surfaces simultaneously.
MXCV	Maximum curvature. This boundary operand constrains each of the curvatures of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also MNCV. This operand controls multiple surfaces simultaneously.
MXEA	Maximum edge thickness air. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be less than the specified target value. See also MXET, MXEG, ETLT, and XXEA. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XXEA for constraining non-rotationally

	symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MXEG	Maximum edge thickness glass. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be less than the target value. See also MXET, MXEA, ETLT, and XXEG. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XXEG for constraining non- rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MXET	Maximum edge thickness. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also "MXEG", "MXEA", "ETLT", and "XXET". This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XXET for constraining non-rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MNSD	Minimum semi-diameter. Constrains the semi-diameter to be larger than the specified target over the surface range between **Surf1** and **Surf2**. This operand controls multiple surfaces simultaneously.
MXSD	Maximum semi-diameter. Constrains the semi-diameter to be less than the specified target over the surface range from **Surf1** to **Surf2**.
TGTH	Sum of glass thicknesses from **Surf1** to **Surf2**. Note that the sum is inclusive, it is not the thickness between the two surfaces. See TTHI.
TTGT	Total thickness greater than. This boundary operand constrains the total thickness, including surface sags of the surface defined by **Surf** and the immediately following surface at their respective semi-diameter values, to be greater than the specified target value. The thickness is calculated along the +y axis if **Code** is zero, the +x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. This operand automatically changes the sign on thicknesses in mirror spaces to always yield a positive value for physically possible lenses. See TTLT and TTVA.
TTHI	Sum of thicknesses of surfaces from **Surf1** to **Surf2**. Note that the sum is inclusive, it is not the thickness between the two surfaces. See TGTH.
TTLT	Total thickness less than. See TTGT.

TTVA	Total thickness value. See TTGT.
XNEA	Minimum edge thickness for the range of air surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are at least the minimum specified thickness. This operand controls multiple surfaces simultaneously. See MNEA. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XNET	Minimum edge thickness for the range of surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are at least the minimum specified thickness. This operand controls multiple surfaces simultaneously. See MNET. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XNEG	Minimum edge thickness for the range of glass surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are at least the minimum specified thickness. This operand controls multiple surfaces simultaneously. See MNEG. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XXEA	Maximum edge thickness for the range of air surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are no more than the maximum specified thickness. This operand controls multiple surfaces simultaneously. See MXEA. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XXEG	Maximum edge thickness for the range of glass surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are no more than the maximum specified thickness. This operand controls multiple surfaces simultaneously. See MXEG. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A

	Zone value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XXET	Maximum edge thickness for the range of surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are no more than the maximum specified thickness. This operand controls multiple surfaces simultaneously. See MXET. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
ZTHI	This operand controls the variation in the total thickness of the range surfaces defined by **Surf1** and **Surf2** over multiple configurations. It is similar to the TTHI operand, except it is an inequality operator. The target value specified is the maximum allowed difference between the TTHI at each defined configuration. For example, if there are 3 configurations where TTHI 3 8 would evaluate to 17, 19, and 18.5, respectively, ZTHI will return 2 (i.e. 19-17) if the target is less than 2. Otherwise, ZTHI returns the target value. To keep all zoom configurations the same length, use a target of 0.

Constraints on lens properties

Operands for constraints on lens properties	
CVOL, MNDT, MXDT, SAGX, SAGY, SSAG, STHI, TMAS, TOTR, VOLU, NORX, NORY, NORZ, NORD, SCUR, SDRV	
NAME	**Description**
CVOL	Cylinder volume. This operand computes the volume in cubic lens units of the smallest cylinder that will contain the range of surfaces defined by **Surf1** and **Surf2**. Only the vertex positions and semi- diameters are used in the calculation, not the sag. The range of surfaces should not include any coordinate breaks.
MNDT	Minimum diameter to thickness ratio. Controls the minimum allowable value on the ratio of surface diameter to center thickness of surfaces from **Surf1** to **Surf2**. Only surfaces with non-unity index of refraction are considered. See also MXDT. This operand controls multiple surfaces simultaneously.
MXDT	Maximum diameter to thickness ratio. Controls the maximum allowable value on the ratio of surface diameter to center thickness from **Surf1** to **Surf2**. Only surfaces with non-unity index of refraction are considered. See also MNDT. This operand controls multiple

	surfaces simultaneously.
SAGX	The sag in lens units of the surface defined by **Surf** at X = the semi-diameter, and Y = 0. See also SSAG.
SAGY	The sag in lens units of the surface defined by **Surf** at Y = the semi-diameter, and X = 0. See also SSAG.
SSAG	The sag in lens units of the surface defined by **Surf** at the coordinate defined by **X** and **Y**. See also SAGX, SAGY.
STHI	Surface Thickness. This operand computes the thickness of the surface defined by **Surf** to the next surface at the coordinate defined by **X** and **Y** on the surface. The calculation accounts for the sag and center thickness of the surface and the sag of the next surface, but not any tilts and decenters between the surfaces.
TMAS	Total mass. Computes the mass of the glass lenses within the range of surfaces from **Surf1** to **Surf2**. The mass of a surface considers the volume enclosed up to the following surface; therefore to compute the mass of a single element the first and last surface numbers should be the same. See "Comments on computing element volumes" for a discussion of how element masses and volumes are computed.
TOTR	Total track (length) of lens in lens units. See "Total track".
VOLU	Volume of element(s) in cubic cm. Computes the volume of the lenses and air spaces for the range of surfaces defined by **Surf1** and **Surf2**. The volume of a surface includes the volume enclosed up to the following surface; therefore to compute the volume of a single element the first and last surface numbers should be the same. See "Comments on computing element volumes" for a discussion of how element masses and volumes are computed.
NORX	Normal vector x component. This operand returns the x component of the surface normal vector at the coordinate defined by **X** and **Y** on the surface defined by **Surf**. If **Global** is zero, the vector is in surface local coordinates, if **Global** is 1, the vector is in global coordinates.
NORY	Normal vector y component. This operand returns the y component of the surface normal vector at the coordinate defined by **X** and **Y** on the surface defined by **Surf**. If **Global** is zero, the vector is in surface local coordinates, if **Global** is 1, the vector is in global coordinates.
NORZ	Normal vector z component. This operand returns the z component of the surface normal vector at any coordinate defined by **X** and **Y** on the surface defined by **Surf**. If **Global** is zero, the vector is in surface local coordinates, if **Global** is 1, the vector is in global

	coordinates.
NORD	Normal distance to the next surface. This operand computes the surface normal vector at the coordinate defined by **X** and **Y** on the surface defined by **Surf**, then returns the distance to the next surface measured along the normal vector.
SCUR	Surface Curvature. Computes curvature and related data of the surface defined by **Surf** at the coordinate defined by **X** and **Y**. In all cases below, the "maximum" data is determined by calculation of the curvature at 50 points equally spaced from the vertex of the surface to the specified **X** and **Y** coordinate, and the largest absolute value found is returned.
	If **data** is 0-3, the returned value is the tangential, sagittal, tangential-sagittal, or maximum tangential-sagittal curvature, respectively.
	If **data** is 4-7, the returned value is the x, y, x-y, or maximum x-y curvature, respectively.
	If **data** is 8-9, the returned value is the absolute value of (R*Sc), where R is the radial coordinate and Sc is the sagittal curvature, or the maximum of this value, respectively.
SDRV	Surface Derivative. Computes the first or second derivative of the surface sag (along the local Z
	axis) of the surface defined by **Surf** at the coordinate defined by **X** and **Y**.
	If **data** is 0 or 1, the returned value is the first derivative in the tangential or sagittal direction.
	If **data** is 2 or 3, the returned value is the second derivative in the tangential or sagittal direction.

Constraints on parameter data

Operands for constraints on parameter data

PMGT, PMLT, PMVA

NAME	Description
PMGT	Parameter greater than. This boundary operand constrains the value of the parameter defined by **Param** for the surface defined by **Surf** to be greater than the target value. The parameter values have different meanings depending upon the surface type. See the Chapter "Surface Types" for a description of the parameter values.
PMLT	Parameter less than. This boundary operand constrains the value of the parameter defined by **Param** for the surface defined by **Surf** to be less than the target value. The parameter values have different meanings depending upon the surface type. See the

	Chapter "Surface Types" for a description of the parameter values.
PMVA	Parameter value. This boundary operand constrains the value of the parameter defined by **Param** for the surface defined by **Surf** to be equal to the target value. The parameter values have different meanings depending upon the surface type. See the Chapter "Surface Types" for a description of the parameter values.

Constraints on extra data

Operands for constraints on extra data

XDGT, XDLT, XDVA

NAME	Description
XDGT	Extra data value greater than. The **Edv#** must be between 1 and 200 to indicate which of the extra data values is selected for the surface defined by **Surf**.
XDLT	Extra data value less than. The **Edv#** must be between 1 and 200 to indicate which of the extra data values is selected for the surface defined by **Surf**.
XDVA	Extra data value. The **Edv#** must be between 1 and 200 to indicate which of the extra data values is selected for the surface defined by **Surf**.

Constraints on glass data

Operands for constraints on glass data

GCOS, GTCE, INDX, MNAB, MNIN, MNPD, MXAB, MXIN, MXPD, RGLA

NAME	Description
GCOS	Glass cost. This operand returns the relative cost factor as listed in the glass catalog for the glass on the surface defined by **Surf**.
GTCE	Glass TCE. This operand returns the Thermal Coefficient of Expansion Alpha1 as listed in the glass catalog for the glass on the surface defined by **Surf**. For non-glass surfaces, see "TCVA".
INDX	Index of refraction. Returns the current index at the surface defined by **Surf** at the wavelength defined by **Wave**.
MNAB	Minimum Abbe number. This boundary operand constrains the Abbe number of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MXAB. This operand controls multiple surfaces simultaneously.
MNIN	Minimum index at d-light. This boundary operand constrains the Nd value of surfaces

	from **Surf1** and **Surf2** to be greater than the specified target value. See also MXIN. This operands controls multiple surfaces simultaneously.
MNPD	See MNPD under "Constraints on lens data"
MXAB	Maximum Abbe number. This boundary operand constrains the Abbe number of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also MNAB. This operand controls multiple surfaces simultaneously.
MXIN	Maximum index at d-light. This boundary operand constrains the Nd value of surfaces between **Surf1** and **Surf2** to be less than the specified target value. See also MNIN. This operand controls multiple surfaces simultaneously.
MXPD	Maximum $\Delta P_{g, F}$. This boundary operand constrains the deviation of the partial dispersion of surfaces between **Surf1** and **Surf2** to be less than the specified target value. See also MNPD. This operand controls multiple surfaces simultaneously.
RGLA	Reasonable glass. This operand restricts the deviation the index, Abbe, and deviation of the partial dispersion values may take from actual glasses in the currently loaded glass catalogs. See "Optimizing zoom and multi-configuration lenses" for a complete discussion including a description of **Wn**, **Wa** and **Wp**. The constraint is active over the surface range specified by **Surf1** and **Surf2**.

Constraints on paraxial ray data

Operands for constraints on paraxial ray data
PANA, PANB, PANC, PARA, PARB, PARC, PARR, PARX, PARY, PARZ, PATX, PATY, YNIP

NAME	**Description**
PANA	Paraxial ray x-direction surface normal at the ray-surface intercept at the wavelength defined by **Wave**. This is the x component of the surface normal vector at the intersection point of the specified paraxial ray and the surface defined by **Surf**, in the local coordinate system. See "Hx, Hy, Px, and Py".
PANB	Paraxial ray y-direction surface normal at the ray-surface intercept at the wavelength defined by **Wave**. This is the y component of the surface normal vector at the intersection point of the specified paraxial ray and the surface defined by **Surf**, in the local coordinate system. See "Hx, Hy, Px, and Py".
PANC	Paraxial ray z-direction surface normal at the ray-surface intercept at the wavelength defined by **Wave**. This is the z component of the surface normal vector at the intersection point of the specified paraxial ray and the surface defined by **Surf**, in the local

	coordinate system. See "Hx, Hy, Px, and Py".
PARA	Paraxial ray x-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PARB	Paraxial ray y-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PARC	Paraxial ray z-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py .
PARR	Paraxial ray radial coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. This is the radial distance from the local axis to the intersection of the surface defined by **Surf** and the specified paraxial ray, in the local coordinate system. See "Hx, Hy, Px, and Py".
PARX	Paraxial ray x-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PARY	Paraxial ray y-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PARZ	Paraxial ray z-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PATX	Paraxial ray x-direction ray tangent. This is the tangent of the angle the paraxial ray makes in the X-Z plane after refraction from surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PATY	Paraxial ray y-direction ray tangent. This is the tangent of the angle the paraxial ray makes in the Y-Z plane after refraction from surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
YNIP	YNI-paraxial. This number is the product of the paraxial marginal ray height times the

	index times the angle of incidence at the surface defined by **Surf** at the wavelength defined by **Wave**. This quantity is related to the narcissus contribution of the specified surface. See Applied Optics, Vol. **21**, 18, p3393.

Constraints on real ray data

Operands for constraints on real ray data

CEHX, CEHY, CENX, CENY, CNAX, CNAY, CNPX, CNPY, DXDX, DXDY, DYDX, DYDY, HHCN, IMAE, MNRE, MNRI, MXRE, MXRI, OPTH, PLEN, RAED, RAEN, RAGA, RAGB, RAGC, RAGX, RAGY, RAGZ, RAID, RAIN, RANG, REAA, REAB, REAC, REAR, REAX, REAY, REAZ, RENA, RENB, RENC, RETX, RETY

NAME	Description
CEHX	Huygens PSF centroid X position. This operand uses the Huygens PSF to determine the x coordinate of the centroid for any field point. The centroid accounts for all apodization and apertures, and optionally polarization. The parameters for this operand are: **Wave**: The wavelength number to use (0 for polychromatic, otherwise use the monochromatic wavelength number). **Field:** The integer field number to use. **Pol?:** Set to 0 to ignore polarization and 1 to consider it. **Pupil Samp**: Use 1 for 32x32, 2 for 64x64, etc. **Image Samp**: Use 1 for 32x32, 2 for 64x64, etc. **All Conf?:** Set to 0 to use the current configuration (defined by the last CONF operand preceding this operand), and 1 to sum over all configurations. See "Huygens PSF" for a full discussion of this option. This feature always uses the default image delta. The evaluation surface is always the image surface (see IMSF). When CEHX is followed by a CEHY with identical settings, both are computed with the same ray set to save time. See also CENX, CENY, CNPX, CNPY, CNAX, and CNAY.
CEHY	Huygens PSF centroid Y position. See CEHX.
CENX	Centroid X position. This operand uses a grid of rays to determine the x coordinate of the centroid of all rays from a single field point. The centroid accounts for apodization and apertures, and optionally polarization. The parameters for this operand are:

	Surf: The surface number to use (use 0 for the image surface). **Wave**: The wavelength number to use (0 for polychromatic, otherwise use the monochromatic wavelength number). **Field:** The integer field number to use. **Pol?:** Set to 0 to ignore polarization and 1 to consider it. **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. When CENX is followed by a CENY with identical settings, both are computed with the same ray set to save time. See also CEHX, CEHY, CNPX, CNPY, CNAX, and CNAY.
CENY	Centroid Y position. See CENX.
CNAX	Centroid angular x direction. This operand computes the X angle in radians relative to the local Z axis of the centroid of rays from any field point. The centroid accounts for apodization and apertures, and optionally polarization. **Surf**: The surface number to use (use 0 for the image surface). **Wave**: The wavelength number to use (0 for polychromatic, otherwise use the monochromatic wavelength number). **Hx/Hy:** The normalized field coordinates to use. See "Hx, Hy, Px, and Py". **Pol?:** Set to 0 to ignore polarization and 1 to consider it. **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. When CNAX is followed by a CNAY with identical settings, both are computed with the same ray set to save time. See also CNAY, CNPX, CNPY, CENX, CENY, CEHX, CEHY.
CNAY	Centroid angular y direction. See CNAX.
CNPX	Similar to CNAX, but computes the centroid position rather than angle.
CNPY	Similar to CNAY, but computes the centroid position rather than angle.
DXDX	Derivative of transverse x-aberration with respect to x-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py" .
DXDY	Derivative of transverse x-aberration with respect to y-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".

DYDX	Derivative of transverse y-aberration with respect to x-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**.
	See "Hx, Hy, Px, and Py".
DYDY	Derivative of transverse y-aberration with respect to y-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**.
	See "Hx, Hy, Px, and Py".
HHCN	Test for the hyperhemisphere condition. Zemax traces the specified ray at the wavelength defined by **Wave** to the surface defined by **Surf**, and computes the x, y, and z intercept coordinates. Then, the x and y coordinates only are used in the sag expression for that surface to see what z coordinate results. If the z coordinates are not the same, then HHCN returns 1, otherwise it returns zero. This operand can be used to prevent optimizations from reaching solutions that require hyperhemispheric surface shapes.
	See "Hx, Hy, Px, and Py".
IMAE	Image analysis data. This operand returns fractional efficiency, centroid, or RMS data as computed by the geometric image analysis feature, using whatever the current default settings are, except for "Show" which is always set to spot diagram for this computation, "Surface", which may be selected using **Surf**, "Field", which may be selected using **Field**, and "Field Size", which may be selected using **Field Size**. To use this operand, first define the settings on the geometric image analysis feature as desired, then press Save on the settings box. The operand IMAE will return the desired data as computed by the image analysis feature.
	If **Surf** is 0, the surface specified by the saved settings will be used. If **Surf** is greater than zero, then the data will be computed at the specified surface.
	If **Field** is 0, the field number specified by the saved settings will be used. If **Field** is greater than zero, then the data will be computed at the specified field.
	If **Field Size** is 0, the Field Size specified by the saved settings will be used. If **Field Size** is greater than zero, then the data will be computed using the specified Field Size.
	Data is 0 for efficiency, 1 for X-centroid, 2 for Y-centroid, or 3-5 for X-, Y- or Radial-direction RMS, respectively. All values other than efficiency are in lens units.
	If **Wave** is 0, the wave number specified by the saved settings file will be used. If **Wave** is greater than zero, then the data will be computed using the specified wavelength number.
	See the discussion "Optimizing with the IMAE operand".
MNRE	Minimum real ray angle of exitance. This boundary operand constrains the minimum ray exit angle over a range of surfaces. The angles are measured in degrees between the surface normal and the exiting ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note

	the angle of exitance is always positive. See also MXRE, MNRI, and MXRI. See "Hx, Hy, Px, and Py".
MNRI	Minimum real ray angle of incidence. This boundary operand constrains the minimum ray incidence angle over a range of surfaces. The angles are measured in degrees between the surface normal and the incident ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note the angle of incidence is always positive. See also MNRE, MXRE, and MXRI. See "Hx, Hy, Px, and Py".
MXRE	Maximum real ray angle of exitance. This boundary operand constrains the maximum ray exit angle over a range of surfaces. The angles are measured in degrees between the surface normal and the exiting ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note the angle of exitance is always positive. See also MNRE, MNRI, and MXRI. See "Hx, Hy, Px, and Py".
MXRI	Maximum real ray angle of incidence. This boundary operand constrains the maximum ray incidence angle over a range of surfaces. The angles are measured in degrees between the surface normal and the incident ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note the angle of incidence is always positive. See also MNRE, MNRI, and MXRE. See "Hx, Hy, Px, and Py".
OPTH	Optical path length. This is the distance, in lens units, the specified ray travels to the surface defined by **Surf** at the wavelength defined by **Wave**. The distance is measured from the object for finite conjugates; otherwise the distance is referenced to the first surface. The optical path accounts for the index of refraction of the media, and for phase adding surfaces such as gratings and binary optics. See PLEN. See "Hx, Hy, Px, and Py".
PLEN	Path length. This operand computes the total optical path length (including index of refraction and phase surfaces) between surfaces **Surf1** and **Surf2** for the specified ray, which is always traced at the primary wavelength. PLEN is essentially the difference between two OPTH operands. See OPTH. See "Hx, Hy, Px, and Py".
RAED	Real ray angle of exitance. This is the angle in degrees between the surface normal and the ray after refraction or reflection for the surface defined by **Surf** at the wavelength defined by **Wave**. See also RAID. See "Hx, Hy, Px, and Py".
RAEN	Real ray angle of exitance. This is the cosine of the angle between the surface normal and the ray after refraction or reflection at the surface defined by **Surf** at the wavelength

	defined by **Wave**. See also RAIN. See "Hx, Hy, Px, and Py".
RAGA	Global ray x-direction cosine. This is the direction cosine of the ray in the global coordinate system at the surface defined by **Surf** at the wavelength defined by **Wave**. The origin of the global coordinate system is at the global reference surface. See "Hx, Hy, Px, and Py".
RAGB	Global ray y-direction cosine. See RAGA.
RAGC	Global ray z-direction cosine. See RAGA.
RAGX	Global ray x-coordinate. This is the coordinate in lens units in the global coordinate system at the surface defined by **Surf** at the wavelength defined by **Wave**. The origin of the global coordinate system is at the global reference surface. See "Hx, Hy, Px, and Py".
RAGY	Global ray y-coordinate. See RAGX.
RAGZ	Global ray z-coordinate. See RAGX.
RAID	Real ray angle of incidence. This is the angle in degrees between the surface normal and the incident ray at the surface defined by **Surf** at the wavelength defined by **Wave**. Note the angle of incidence is always positive. See also RAED. See "Hx, Hy, Px, and Py".
RAIN	Real ray angle of incidence. This is the cosine of the angle between the surface normal and the ray before refraction at the surface defined by **Surf** at the wavelength defined by **Wave**. See also RAEN. See "Hx, Hy, Px, and Py".
RANG	Ray angle in radians with respect to z axis. The angle is measured with respect to the local Z axis of the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAA	Real ray x-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py" .
REAB	Real ray y-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".

REAC	Real ray z-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAR	Local real ray radial coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAX	Local real ray x-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAY	Local real ray y-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAZ	Local real ray z-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RENA	Real ray x-direction surface normal at the ray-surface intercept at the surfaced defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RENB	Real ray y-direction surface normal at the ray-surface intercept at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RENC	Real ray z-direction surface normal at the ray-surface intercept at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RETX	Real ray x-direction ray tangent (slope) at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RETY	Real ray y-direction ray tangent (slope) at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".

Constraints on element positions

Operands for constraints on element positions	
GLCA, GLCB, GLCC, GLCR, GLCX, GLCY, GLCZ	
NAME	**Description**
GLCA	Global x-direction orientation vector component of the surface defined by **Surf**.
GLCB	Global y-direction orientation vector component of the surface defined by **Surf**.
GLCC	Global z-direction orientation vector component of the surface defined by **Surf**.
GLCR	Global Coordinate Rotation Matrix component at the surface defined by **Surf**. The 3 x 3 R matrix has 9 components. If **Data** is 1, GLCR returns R[1][1], if **Data** is 2, GLCR returns R[1][2], etc... through **Data** = 9 returning R[3][3].
GLCX	Global vertex x-coordinate of the surface defined by **Surf**.
GLCY	Global vertex y-coordinate of the surface defined by **Surf**.
GLCZ	Global vertex z-coordinate of the surface defined by **Surf**.

Changing system data

Operands for changing system data	
CONF, IMSF, PRIM, SVIG, WLEN, CVIG, FDMO, FDRE	
NAME	**Description**
CONF	Configuration. This operand is used to change the configuration number to the configuration defined by **Cfg#** during merit function evaluation. This permits optimization across multiple configurations with a single merit function. This operand does not use the target or weight columns.
IMSF	Image Surface. This operand dynamically changes which surface is interpreted as the "image surface" for all subsequent operands. The new image surface is defined by **Surface**. The primary usage for this operand is to optimize image quality at intermediate surfaces. Set **Surface** to 0 to restore the image surface back to the original surface.
	If **Refocus?** is 0, the specified surface becomes the new image surface and is not modified in any way. If **Refocus?** is 1, then a dummy plane surface will follow the specified surface, and a paraxial focus solve will be used to place this dummy surface at the focus of the specified surface. This option is most useful for optimizing aberrations at a virtual focus for intermediate surfaces that are not already at focus.
	Note this operand only temporarily changes a copy of the lens data use for evaluating the merit function and has no impact on the original lens data. Care should be taken

	when the selected surface precedes the original stop surface. For information on how this operand is implemented see "Evaluating results at intermediate surfaces".
PRIM	Primary wavelength. This is used to change the primary wavelength number to the wavelength defined by **Wave** during merit function evaluation. This operand does not use the target or weight columns.
SVIG	Sets the vignetting factors for the current configuration. The **Precision** is either 0, 1, or 2; for high, medium, or low precision, respectively. The SVIG operand can take a significant time to execute at high precision, although the resulting optimization is generally superior. Note that the operands SVIG and CVIG only modify the vignetting factors for evaluating subsequent optimization operands. When the end of the merit function is reached, the vignetting factors will be restored to their original values. See also "CVIG".
WLEN	Wavelength. This operand returns the wavelength defined by **Wave** in micrometers.
CVIG	Clears the vignetting factors. This operand clears the current vignetting factors for the remainder of the operands in the current configuration. See also "SVIG".
FDMO	Field Data Modify. This operand allows temporary modification of field position data. The operand allows new field coordinates and vignetting factors (except for vignetting angle, which cannot be modified by this operand) for any field coordinate. All subsequent operands will use the modified field data. The original field data is restored when a FDRE operand is reached with the same field number argument, or a CONF operand is reached (regardless of the configuration number referenced by the CONF operand), or after the last operand in the merit function is reached. **Field**: The field number to modify. **Hx, Hy**: The normalized field coordinates for the new field position. **VDX, VDY, VCX, VCY**: The vignetting factor data for the new field position. See "Vignetting factors".
FDRE	Field Data Restore. See FDMO. **Field**: The field number to restore.

General math operands

General math operands
ABSO, ACOS, ASIN, ATAN, CONS, COSI, DIFF, DIVB, DIVI, EQUA, LOGE, LOGT, MAXX, MINN, OPGT, OPLT, OPVA, OSUM, PROB, PROD, QSUM, RECI, SQRT, SUMM, SINE, TANG, ABGT, ABLT
NAME **Description**

ABSO	Absolute value of the operand defined by **Op#**.
ACOS	Arc cosine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
ASIN	Arcsine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
ATAN	Arctangent of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
CONS	Constant value. This is used to enter in constant values for use in other operand computations. The value will be identical to the target value.
COSI	Cosine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
DIFF	Difference of two operands (**Op#1** - **Op#2**).
DIVB	Divides the value of any prior operand defined by **Op#** by any factor defined by **Factor**.
DIVI	Division of first by second operand (**Op#1** / **Op#2**). See also "RECI".
EQUA	Equal operand. This operand constrains all operands within the range of operands defined by **Op#1** and **Op#2** to have the same value within the tolerance specified by the target. The value of this operand is computed by finding the average of the range of values, and then summing the absolute value of the errors between each operand and the average if the error exceeds the target value. See SUMM and OSUM.
LOGE	Log base e of an operand. **Op#** is the row number of the operand value to take the log of. If the value is less than or equal to zero, the value zero is returned.
LOGT	Log base 10 of an operand. **Op#** is the row number of the operand value to take the log of. If the value is less than or equal to zero, the value zero is returned.
MAXX	Returns the largest value within the indicated range of operands defined by **Op#1** and **Op#2.** See MINN.
MINN	Returns the smallest value within the indicated range of operands. See MAXX.
OPGT	Operand greater than. This is used to make the value of the operand defined by **Op#**

	greater than the target value.
OPLT	Operand less than. This is used to make the value of the operand defined by **Op#** less than the target value.
OPVA	Operand value. This operand constrains the value of the operand defined by **Op#** to be equal to the target value.
OSUM	Sums the values of all operands between the two operands defined by **Op#1** and **Op#2**. See SUMM.
PROB	Multiplies the value of the operand defined by **Op#** by the factor defined by **Factor**.
PROD	Product of two operands (**Op#1** X **Op#2**). See PROB.
QSUM	Quadratic sum. This operand squares and then adds all operands between the two operands defined by **Op#1** and **Op#2**, then takes the square root of the sum. See also SUMM, OSUM, EQUA.
RECI	Returns the reciprocal of the value of operand **Op#1**. See also "DIVI".
SQRT	Square root of the operand defined by **Op#**.
SUMM	Sum of two operands (**Op#1** + **Op#2**). See OSUM.
SINE	Sine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
TANG	Tangent of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
ABGT	Absolute value of operand greater than. This is used to make the absolute value of the operand defined by **Op#** greater than the target value.
ABLT	Absolute value of operand less than. This is used to make the absolute value of the operand defined by **Op#** less than the target value.

Multi-configuration (zoom) data

Operands for multi-configuration (zoom) data

CONF, MCOL, MCOG, MCOV, ZTHI

NAME	Description
CONF	See CONF under "Changing system data" operands
MCOG	Multi-configuration operand greater than. This is used to constrain values in the multi-configuration editor. **Op#** defines which multi-configuration operand is used. **Cfg#** defines which configuration is used.
MCOL	Multi-configuration operand less than. This is used to constrain values in the multi-configuration editor. See MCOG.
MCOV	Multi-configuration operand value. This is used to directly target or compute values in the multi- configuration editor. See MCOG.
ZTHI	See ZTHI under "Constraints on lens data" operands

Gaussian beam data

Operands for Gaussian beam data

GBPD, GBPP, GBPR, GBPS, GBPW, GBPZ, GBSD, GBSP, GBSR, GBSS, GBSW

NAME	Description
GBPD	Gaussian beam (paraxial) divergence in the optical space following the surface defined by **Surf** at the wavelength defined by **Wave**. The other parameters are: **UseX**: If this parameter is non-zero, then the computation is for the x-direction beam, otherwise, it is for the y-direction. **W0**: The input beam waist size in lens units. **S1toW**: The distance from Surface 1 to the waist location in lens units. See the Gaussian beam feature for details. **M2 Factor**: The M Squared value for the beam. See the Gaussian beam feature for details.
GBPP	Gaussian beam (paraxial) position, which is the distance from the waist to the surface, in the optical space following the specified surface. See GBPD.

GBPR	Gaussian beam (paraxial) radius of curvature in the optical space following the specified surface. See GBPD.
GBPS	Gaussian beam (paraxial) size in the optical space following the specified surface. See GBPD.
GBPW	Gaussian beam (paraxial) waist in the optical space following the specified surface. See GBPD.
GBPZ	Gaussian beam (paraxial) Rayleigh range in the optical space following the specified surface. See GBPD.
GBSD	Gaussian beam (skew) divergence in the optical space following the specified surface. The input beam is aligned along the chief ray of the field defined by **Field**. The other parameters are: **In#**: The surface number to start the propagation. **Out#**: The surface at which to compute the Skew Gaussian Beam data. **Wave**: The wavelength number to use. **StoW**: The starting surface to waist distance in lens units. **W0**: The input beam waist size in lens units. If **W0** is positive, then the computation is for the y- direction beam, otherwise, it is for the x-direction. For details on the Skew Gaussian Beam feature, see "Skew Gaussian Beam".
GBSP	Gaussian beam (skew) position, which is the distance from the waist to the surface, in the optical space following the specified surface. See GBSD.
GBSR	Gaussian beam (skew) radius in the optical space following the specified surface. See GBSD.
GBSS	Gaussian beam (skew) size in the optical space following the specified surface. See GBSD.
GBSW	Gaussian beam (skew) waist in the optical space following the specified surface. See GBSD.

Gradient index control operands

Gradient index control operands

DLTN, GRMN, GRMX, InGT, InLT, InVA, LPTD

NAME	Description
DLTN	Delta N. Computes the difference between the maximum and minimum index of refraction on axis for a gradient index surface defined by **Surf**. The wavelength used is defined by **Wave**. The minimum and maximum z coordinates used are computed by accounting for the sag of both ends of the surface. See the section "Using gradient index operands".
GRMN	Gradient index minimum index. This boundary operand sets the minimum allowable index of refraction value for the gradient index surface defined by **Surf** at the wavelength defined by **Wave**. The index is checked at six places: the front vertex, front +y top, front +x side, rear vertex, rear +y top, and the rear +x side. See also InGT, InLT and GRMX.
GRMX	Gradient index maximum index. This boundary operand sets the maximum allowable index of refraction value for the gradient index surface defined by **Surf** at the wavelength defined by **Wave**. The index is checked at six places: the front vertex, front +y top, front +x side, rear vertex, rear +y top, and the rear +x side. See also InGT, InLT and GRMN.
InGT	Index "n" greater than. This boundary operand constrains the index of refraction at the wavelength defined by **Wave** of the gradient index surface defined by **Surf** at one of six points inside the gradient index lens. For n=1, the point is the front vertex, n=2 is the front +y top, n=3 is the front +x side, n=4 is the rear vertex, n=5 is the rear +y top, and n=6 is the rear +x side. In all cases the operand bounds the index at the specified point to be greater than the specified target value. For example, "I4GT" constrains the minimum index at the rear vertex of the surface (i.e. the vertex of the next surface) of the gradient index lens. In all cases the +y top and +x side distance is defined by the larger of the front and rear semi-diameters set on the main spreadsheet. See also GRMN and GRMX, which are similar operands that are easier to use.
InLT	Index "n" less than. This operand is similar to InGT except it constrains the maximum value of the index of refraction rather than the minimum. See InGT for a complete description of the parameter "n".
InVA	This operand is similar to InGT except it constrains the current value of the index of refraction. See InGT for a complete description of the parameter "n".
LPTD	This boundary operand constrains the slope of the axial gradient index profile from changing signs within a gradient index component for the surface defined by **Surf**. See the section "Using gradient index operands".

Foucault analysis

Operands for Foucault analysis

FOUC	
NAME	**Description**
FOUC	Foucault analysis. This operand returns the RMS difference between the computed and reference shadowgram as computed by the Foucault analysis feature, using whatever the current default settings are. To use this operand, first define the settings on the Foucault analysis feature as desired, then press Save on the settings box. The operand FOUC will return the RMS difference between the computed and reference shadowgrams, regardless of how the "data" option is set in the saved configuration file. Using this operand, the optical system wavefront aberrations may be optimized to produce the reference shadowgram.

Ghost focus control

Operands for ghost focus control	
GPIM, GPRT, GPRX, GPRY, GPSX, GPSY	
NAME	**Description**
GPIM	Ghost pupil image. GPIM controls the location of ghost pupils (and optionally ghost images) relative to the image surface. Double-bounce ghosts form images of the pupil, and if these images are formed near the focal plane they will contaminate the image with unwanted light. This is the cause of the familiar "sun flare" images of the pupil seen through camera lenses pointed near the sun. The operand computes any one specific or all possible ghost pupil image locations and returns one over the absolute value of the distance from the image surface to the closest pupil ghost. The operand is defined in this manner so it can be targeted to zero and weighted and optimized to reduce ghost pupil affects. If the **Surf1** and **Surf2** parameters are set to specific surface numbers, that specific ghost path is computed, if either or both of the **Surf1** and **Surf2** values are -1, then all possible surface combinations are considered. For example, if **Surf1** is 12 and **Surf2** is -1, then all double bounces that first bounce off surface 12 and then 11, 10, 9, etc. are considered. If both numbers are negative, all possible ghosts are considered. This same operand also can be used for detecting and controlling image ghosts (which are distinct from pupil ghosts) by changing **Mode** from 0 to 1, or to control ghost pupil magnification (the ratio of the ghost exit pupil diameter to entrance pupil diameter), by setting **Mode** to 2. The **WFB** and **WSB** columns will list the worst combination found for reference and possible further analysis. Only surfaces with index changes are considered as possible ghost generators. First bounces off mirrors are ignored. See also GPRT, GPRX, GPRY, GPSX, and GPSY.
GPRT	Ghost ray transmission. This operand computes the unpolarized transmission of any ray from object to image surface for a double-bounce ghost path. The **Surf1** and **Surf2** values define the first and second surface numbers to define the ghost path. Note the

	first bounce surface number must be larger than the second bounce surface number. The primary wavelength is always used.
	GPRT makes no changes to the field or aperture specifications, and therefore may return meaningless results if the field or aperture data are defined in a way that changes the ghost path. No warning is issued in these cases. For example, using GPRT while using image height for field points or using image space F/# for the system aperture type will likely return useless results. The recommended approach is to use field angles or object heights, use entrance pupil diameter, and place the stop prior to the first bounce surface. To study the double bounce system in detail see "Ghost Focus Generator".
	See "Hx, Hy, Px, and Py.
GPRX	Ghost ray real x coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the x coordinate of the real ray on the image surface.
GPRY	Ghost ray real y coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the y coordinate of the real ray on the image surface.
GPSX	Ghost ray paraxial x coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the x coordinate of the paraxial ray on the image surface.
GPSY	Ghost ray paraxial y coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the y coordinate of the paraxial ray on the image surface.

Fiber coupling operands

Fiber coupling operands	
FICL, FICP, POPD	
NAME	**Description**
FICL	Fiber coupling efficiency for single mode fibers. The calculated value is the total coupled energy efficiency, relative to unity. The parameters of this operand are:
	Samp: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc.
	Wave: The wavelength number to use.
	Field: The integer field position number.
	IgSrc: If this parameter is zero, then the object source fiber is considered; otherwise, the object source fiber is ignored.

	Sna: The source fiber NA.
	Rna: The receiver fiber NA.
	Data: This controls what data is returned as well as the algorithm to use. If **Data** is 0, the fast algorithm is used and the power coupling is returned. If **Data** is 1, the Huygens algorithm is used and the power coupling is returned. If **Data** is 2 or 4, the fast algorithm is used and the real or imaginary part of the amplitude coupling is returned, respectively. If **Data** is 3 or 5, the Huygens algorithm is used and the real or imaginary part of the amplitude coupling is returned, respectively.
	Pol?: Set to 0 to ignore polarization and 1 to consider it.
	See "Fiber Coupling Efficiency" for details. See also FICP.
FICP	Fiber coupling as computed using the Physical Optics Propagation (POP) algorithm, using whatever the current default settings are for the POP feature.
	Data: This controls what data is returned. If **Data** is 0 the total power coupling is returned. If **Data** is 1, the amplitude coupling for the Ex field is returned. If **Data** is 2 or 3, the real or imaginary part of the amplitude coupling for the Ex field is returned, respectively. If **Data** is 4, the amplitude coupling for the Ey field is returned. If **Data** is 5 or 6, the real or imaginary part of the amplitude coupling for the Ey field is returned, respectively. If the beam is unpolarized the Ey values wil be zero.
	To use this operand, first define the settings on the POP analysis feature as desired, then press Save on the settings box. The operand FICP will return the same efficiency as computed by the POP feature.
	This operand is redundant with the more general POPD.
	See "Computing Fiber Coupling". See also FICL.
POPD	Physical Optics Propagation Data. For important details see "PHYSICAL OPTICS PROPAGATION". To use this operand, first define the settings on the POP analysis feature as desired, then press Save on the settings box. The operand will return data based upon the selected settings.
	If **Surf** is zero, then the saved ending surface number will be used; otherwise, the specified surface will be used as the ending surface. If **Wave** is zero, then the saved wavelength number will be used; otherwise, the specified wavelength number will be used. If **Field** is zero, then the saved field number will be used; otherwise, the specified field number will be used.
	Data determines what data the POP feature will compute and return as follows:
	0: The total fiber coupling. This is the product of the system efficiency and the receiver efficiency.
	1: The system efficiency for fiber coupling.
	2: The receiver efficiency for fiber coupling.

| | 3: The total power.

4: The peak irradiance.

5, 6, 7: The pilot beam position, Rayleigh range, beam waist (x).

8, 9, 10: The pilot beam position, Rayleigh range, beam waist (y).

11, 12, 13: The local X, Y, Z coordinates of the center of the beam array on the end surface (this is a reference point and is not related to the amplitude of the beam).

21, 22: The X, Y coordinates of the centroid of the intensity distribution in local coordinates relative to the center of the beam.

23, 24, 25, 26: The X, Y beam width and X, Y M-squared values, respectively. See "Beam width and M-squared".

27, 28, 29: The second moment of the beam for x^2, y^2, and xy respectively.

30, 31, 32: The mean, RMS and PTV irradiance variation, respectively, of the non-zero amplitude portion of the beam. These operands should only be used when the beam has nearly uniform irradiance and has just been clipped by a surface aperture.

33, 34, 35: The mean, RMS and PTV phase variation in radians, respectively, of the non-zero amplitude portion of the beam. These operands should only be used when the beam has nearly uniform irradiance and has just been clipped by a surface aperture.

40, 41, 42: The fraction of the total power enclosed within a circle of radius specified by the **Xtr1** value in lens units, referenced to the beam centroid (40), chief ray (41), or surface vertex (42).

50, 51, 52: The radius in lens units of the circle at which the fraction of the total power enclosed is equal to the value specified by the **Xtr1** value. The circle is centered on the beam centroid (50), chief ray (51), or surface vertex (52).

60, 61, 62, 63: The fiber coupling receiver efficiency amplitude and phase in radians for the Ex field (60 and 61) and the Ey field (62 and 63). These values do not consider system efficiency (see **Data** type 1 above).

Undefined **Data** values will return 0.

The **Xtr1** and **Xtr2** values are only used by selected data numbers which are reserved for future expansion of this feature.

If adjacent POPD operands all have the same **Surf**, **Wave**, **Field**, **Xtr1**, and **Xtr2** values, then the POP analysis is done just once and all data returned at one time. Note the POPD operands must be on adjacent rows in the MFE for this efficiency to be implemented. |

Relative illumination operand

Relative illumination operands	
RELI, EFNO	
NAME	**Description**
RELI	Relative illumination. This operand computes the relative illumination of the field point defined by **Field** relative to the (0, 0) field point. Note that in some systems, the illumination increases off axis, and for these systems the RI may be greater than one since the RELI operand uses (0, 0) as a reference field point. Vignetting factors are always removed for this calculation. The other parameters are: **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. **Wave**: The wavelength number to use. **Pol?**: Set to 0 to ignore polarization and 1 to consider it. See also EFNO.
EFNO	Effective F/#. This operand computes the Effective F/# of the field point defined by **Field**. For a discussion of Effective F/# see "Effective F/#". The other parameters are: **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. **Wave**: The wavelength number to use. **Pol?**: Set to 0 to ignore polarization and 1 to consider it. See also RELI.

Optimization with ZPL macros

Operands for optimization with ZPL macros	
ZPLM	
NAME	**Description**
ZPLM	Used for optimizing numerical results computed in ZPL macros. See "User defined operands". See also UDOP.

User defined operands

User defined operands	
UDOP	
NAME	**Description**
UDOP	User defined operand. Used for optimizing numerical results computed in externally

compiled programs. See "User defined operands". See also ZPLM.

Merit function control operands

Merit function control operands	
BLNK, DMFS, ENDX, GOTO, OOFF, SKIN, SKIS, USYM	
NAME	**Description**
BLNK	Does nothing. Used for separating portions of the operand list. A comment line may optionally be typed in the space to the right of the operand name; this comment will be displayed in the editor as well as in the merit function listing.
DMFS	Default merit function start. This operand is a flag to indicate where the default merit function should be appended, if one is subsequently created. The row number after this operand will appear as the default "Start At" row on the default merit function dialog box.
ENDX	End execution. Terminates the computation of the merit function. All remaining operands are ignored.
GOTO	Skips all operands between the GOTO operand line and the operand number defined by **Op#**. Execution of the merit function will start again at the **Op#** line.
OOFF	This operand indicates an unused entry in the operand list. OOFF operands are automatically converted to BLNK operands upon evaluation of the merit function. OOFF is only used to indicate that the merit function operand type was not recognized.
SKIN	Skip if not symmetric. See SKIS.
SKIS	Skip if symmetric. If the lens is rotationally symmetric, then computation of the merit function continues at the operand defined by **Op#**.
USYM	If present in the merit function, this operand instructs Zemax to assume radial symmetry exists in the lens even if Zemax detects symmetry does not exist. This speeds execution of the merit function in some special cases. See "Assume Axial Symmetry".

Constraints on non-sequential object data

Operands for constraints on non-sequential object data
FREZ, NPGT, NPLT, NPVA, NPXG, NPXL, NPXV, NPYG, NPYL, NPYV, NPZG, NPZL, NPZV, NSRM, NTXG, NTXL, NTXV, NTYG, NTYL, NTYV, NTZG, NTZL, NTZV

NAME	Description
FREZ	Freeform Z object constraints. This operand computes various values from the shape of a Freeform
	Z object described in "Freeform Z". **Surf** is the surface number of the NSC group, use
	1 for NSC program mode. **Object** is the object number, which must be a Freeform Z object for this
	operand to compute any data. The **Data** value determines what value is computed by the operand
	as follows:
	1: Maximum z value. Nnote minimum z value is always zero.
	2: Maximum z increment. This is the biggest difference between adjacent z control points.
	3: Minimum z increment. This is the smallest difference between adjacent z control points.
	4: Minimum y value. This value is for the entire length of the solid, not just at control points.
	5: Maximum y value. This value is for the entire length of the solid, not just at control points.
	6: Maximum y increment. This is the biggest difference between adjacent y control points.
	7: Minimum y increment. This is the smallest difference between adjacent y control points.
	8: Volume of the object in lens units cubed.
	9: Monotonic deviation. This is the largest change in adjacent y values whose sign deviates from the sign of the first y change going from 0 to positive z coordinates.
	10: Minimum slope.
	11: Maximum slope.
	Once the **Data** has been computed, the **Mode** value determines how the operand behaves. If **Mode** is 1, the FREZ operand returns the data. If **Mode** is 2, the operand returns the data value only if the data value is less than the target value, otherwise the target value is returned. This mode is used to enforce a "greater than" boundary on the data. If **Mode** is 3, the operand returns the data value only if the data value is greater than the target value, otherwise the target value is returned. This mode is used to enforce a "less than" boundary on the data.

NPGT	Non-sequential parameter greater than. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Object** defines the object number in the NSC group. The parameter number is defined by **Param**.
NPLT	Non-sequential parameter less than. See NPGT.
NPVA	Non-sequential parameter value. See NPGT.
NPXG	Non-sequential object position x greater than. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Object** defines the object number in the NSC group. If **Ref?** is 0, the coordinates are relative to the reference object. If **Ref?** is 1, the coordinates are relative to the origin of the NSC coordinate system or entry port. If **Ref?** is 2, then the coordinates are relative to the global coordinate reference surface.
NPXL	Non-sequential object position x less than. See NPXG.
NPXV	Non-sequential object position x value. See NPXG.
NPYG	Non-sequential object position y greater than. See NPXG.
NPYL	Non-sequential object position y less than. See NPXG.
NPYV	Non-sequential object position y value. See NPXG.
NPZG	Non-sequential object position z greater than. See NPXG.
NPZL	Non-sequential object position z less than. See NPXG.
NPZV	Non-sequential object position z value. See NPXG.
NSRM	Non-sequential Rotation Matrix component. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Object** defines the object number in the NSC group. If **Ref?** is 0, the coordinates are relative to the reference object. The rotation matrix will always be the identity matrix when using this reference. If **Ref?** is 1, the coordinates are relative to the origin of the NSC coordinate system or entry port. If **Ref?** is 2, then the coordinates are relative to the global coordinate

	reference surface.
	The 3 x 3 R matrix has 9 components. If **Data** is 1, NSRM returns R[1][1], if **Data** is 2, NSRM returns
	R[1][2], etc... through **Data** = 9 returning R[3][3].
NTXG	Non-sequential object tilt about x greater than. See NPXG.
NTXL	Non-sequential object tilt about x less than. See NPXG.
NTXV	Non-sequential object tilt about x value. See NPXG.
NTYG	Non-sequential object tilt about y greater than. See NPXG.
NTYL	Non-sequential object tilt about y less than. See NPXG.
NTYV	Non-sequential object tilt about y value. See NPXG.
NTZG	Non-sequential object tilt about z greater than. See NPXG.
NTZL	Non-sequential object tilt about z less than. See NPXG.
NTZV	Non-sequential object tilt about z value. See NPXG.

Non-sequential ray tracing and detector operands

Non-sequential ray tracing and detector operands	
NSDC, NSDD, NSDE, NSDP, NSLT, NSRA, NSRM, NSRW, NSST, NSTR, NSTW, REVR	
NAME	**Description**
NSDC	Non-sequential coherent data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector.
	If **Pix#** is a positive integer, then the data from the specified pixel is returned. If **Pix#** is zero, then the sum of the data for all pixels for that detector object is returned.

	Data is 0 for real, 1 for imaginary, 2 for amplitude, and 3 for power. See "User defined operands" for complete details.
NSDD	Non-sequential incoherent intensity data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector. If **Det#** is zero, then all detectors are cleared. If **Det#** is less than zero, then the detector defined by the absolute value of **Det#** only is cleared. Note that one NSDD operand clearing *all* detectors must be present prior to any subsequently defined NSDD operands.

For Detector Rectangles, Detector Surfaces, and all faceted detectors: If **Pix#** is a positive integer greater than zero, then the data from the specified pixel is returned. Otherwise, the following data is returned depending upon the value of **Pix#**:

0: Sum of the total flux in position space, average flux/area in position space, or total flux in angle space for all pixels for that detector object, for **Data** = 0, 1, or 2, respectively.

-1: Maximum flux or flux/area.

-2: Minimum flux or flux/area.

-3: Number of rays striking the detector.

-4: Standard deviation (RMS from the mean) of all the *non-zero* pixel data.

-5: The mean value of all the *non-zero* pixel data.

-6, -7, -8: The x, y, or z coordinate of the position or angle Irradiance or Intensity centroid, respectively.

-9,-10, -11, -12, -13: The RMS radius, x, y, z, or xy cross term distance or angle of all the pixel data with respect to the centroid. These are the second moments r^2, x^2, y^2, z^2, and xy, respectively.

Data is 0 for flux, 1 for flux/area, 2 for flux/solid angle pixel, and 3 for normalized flux. Only values 0 and 1 (for flux and flux/area) are supported for faceted detectors. A value of 3 is only supported when **Pix#** is a positive integer; in this case the returned flux for the pixel is normalized to the peak flux for all consecutive NSDD operands in the Merit Function Editor which refer to the same surface and detector. A value of 3 should generally be used only as a part of the NSC Bitmap Merit Function (see "NSC Bitmap Merit Function Tool").

For Detector Rectangles only: If **Pix#** is not a positive integer, then a number of edge pixels may be ignored in the calculation if desired. The number of edge pixels that will be ignored is given by the **# Ignored** input. For non-zero values of **# Ignored**, the number of pixels accounted for in the calculation will be reduced along all boundaries of the detector. For example, if a detector has 100 pixels in X and 200 pixels in Y and the **# Ignored** value is 2, then the calculation will be based upon detector data from 96 pixels in X and 196 pixels in Y, with data from the last 2 edge pixels along the left, right, bottom and top boundaries all ignored. The only calculation not affected by the **# Ignored** input is the number of rays striking the detector, as given by **Pix #** = -3 (in addition to all values calculated for positive integer inputs to **Pix #**).

For Detector Volumes: **Pix#** is interpreted as the voxel number. For **Data** values of 0, 1, or 2, the returned value is incident flux, absorbed flux, or absorbed flux per unit volume, respectively. If **Pix#** is zero, the value returned is the sum for all pixels. |

	For Detector Color objects, use NSDE instead. For Detector Polar objects, use NSDP.
NSDE	Non-sequential Detector Color object data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector. If **Det#** is zero, then all detectors are cleared. If **Det#** is less than zero, then the detector defined by the absolute value of **Det#** only is cleared.

If **Pix#** is a positive integer greater than zero, then the data from the specified pixel is returned. Otherwise, the following data is returned depending upon the value of **Pix#**: 0: Sum of all data for all pixels on the detector. For chromaticity coordinates this is the average over all pixels in the detector that have non-zero lumen values. If **Pix#** is zero the \<area\> values described below are the area of the entire detector, measured in analysis units for position space data or solid angle in steradians for angle space data.

-1: Maximum data.

-2: Minimum data.

-3: Number of rays striking the detector.

Angle? is 0 for position space data, 1 for angle space data.

Data is an integer code that defines the type of data to return, defined as follows:

1: power in \<sprefix\> watts where \<sprefix\> is the source units prefix.

2: power/area in \<aprefix\> watts per \<area\> where \<aprefix\> is the analysis units prefix. If **Angle?**

is 0 \<area\> is the area in analysis units. If **Angle?** is 1 then \<area\> is solid angle in steradians.

3: power in \<sprefix\> lumens where \<sprefix\> is the source units prefix.

4: power/area in \<aprefix\> lumens per \<area\> where \<aprefix\> is the analysis units prefix. If **Angle?** is 0 \<area\> is the area in analysis units. If **Angle?** is 1 then \<area\> is solid angle in steradians.

5/6: CIE 1931 chromaticity coordinate x/y.

7/8: CIE 1976 chromaticity coordinate u'/v'.

9/10/11: CIE 1931 tristimulus X/Y/Z value in lumens per \<area\>. If **Angle?** is 0 \<area\> is the area in analysis units. If **Angle?** is 1 then \<area\> is solid angle in steradians.

12/13: Color Rendering Index (CRI) and Correlated Color Temperature (CCT), the latter value in Kelvin. These values are only returned if the option to "Record Spectral Data" has been selected for the Detector Color object (only available in Zemax-Premium) and if the **Pix#** and **Angle?** are both zero. The returned values are based on the variation of flux with wavelength over the entire detector.

14/15/16: Normalized CIE 1931 tristimulus X/Y/Z value. Identical to the values returned by Data 9, 10, and 11, but normalized to the peak tristimulus Y value for all consecutive NSDE operands in the Merit Function Editor which refer to the same surface and detector. Generally used only as a part of the NSC Bitmap Merit Function (see "NSC Bitmap Merit |

	Function Tool"). Only valid if **Pix#** is a positive integer and **Angle?** is zero.
	If **Pix#** is not a positive integer, then a number of edge pixels may be ignored in the calculation if desired. The number of edge pixels that will be ignored is given by the **# Ignored** input. For non- zero values of **# Ignored**, the number of pixels accounted for in the calculation will be reduced along all boundaries of the detector. For example, if a detector has 100 pixels in X and 200 pixels in Y and the **# Ignored** value is 2, then the calculation will be based upon detector data from 96 pixels in X and 196 pixels in Y, with data from the last 2 edge pixels along the left, right, bottom and top boundaries all ignored. Calculations that are not affected by the **# Ignored** input are the number of rays striking the detector, as given by **Pix #** = -3, and the CRI and CCT values, as given by Data = 12 and Data = 13, respectively (the **# Ignored** input also does not affect any values calculated for positive integer inputs to **Pix #**).
	If the source units are joules, then the data 1 and 2 are in joules and data 3 and 4 are in talbots. See "Units". See also NSDD and NSDP.
NSDP	Non-sequential Detector Polar object data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector. If **Det#** is zero, then all detectors are cleared. If **Det#** is less than zero, then the detector defined by the absolute value of **Det#** only is cleared.
	If **Pix#** is a positive integer greater than zero, then the data from the specified pixel is returned. Otherwise, the following data is returned depending upon the value of **Pix#**:
	0: Sum of all data for all pixels on the detector. For chromaticity coordinates this is the average over all pixels in the detector that have non-zero lumen values.
	-1: Maximum data.
	-2: Minimum data.
	-3: Number of rays striking the detector.
	-4: The radial RMS in degrees of the distribution with respect to 0 degrees. If **Pix#** is set equal to -
	4, then the only valid inputs for **Data** are 1 or 3. All other inputs will return a value of zero.
	Data is an integer code that defines the type of data to return, defined as follows:
	1: power in <sprefix> watts where <sprefix> is the source units prefix.
	2: power/solid angle in <sprefix> watts per steradian where <sprefix> is the source units prefix.
	3: power in <sprefix> lumen where <sprefix> is the source units prefix.
	4: power/solid angle in <sprefix> lumens per steradian where <sprefix> is the source units prefix.
	5/6: CIE 1931 chromaticity coordinate x/y.

	7/8: CIE 1976 chromaticity coordinate u'/v'.
	9/10/11: CIE 1931 tristimulus X/Y/Z value in lumens/steradian.
	If the source units are joules, then the data 1 and 2 are in joules and data 3 and 4 are in talbots. See "Units". See also NSDD and NSDE.
NSLT	Non-sequential LightningTrace. This operand traces mesh rays as defined by the LightningTrace analysis. **Surf** is the surface number of the Non-Sequential surface. **Src#** refers to the object number of the desired source from which to trace mesh rays. If **Src#** is zero, rays will be traced from all sources. **Ray Samp** defines the sampling to use in defining the ray mesh: 0 = Low (1X), 1
	= 4X, 2 = 16X, 3 = 64X, 4 = 256X, 5 = 1024X. **Edge Samp** defines the sampling to use for resolving the ray mesh around the edges of objects, with the same interpretation for the numeric inputs as for **Ray Samp** (but with inputs only going from 0 to 4). **RT?** defines whether mesh rays should be traced (when **RT?** = 0) or whether all analysis rays should be traced using conventional ray tracing (when **RT?** = 1). If **RT?** = 1, then this operand will act just like the NSTR operand with polarization, splitting, and scattering all turned off.
	An NSDD, NSDE, or NSDP operand with Det# set equal to 0 must be included in the merit function prior to this operand.
	See "The LightningTrace Control" for more information. See also NSTR.
NSRA	Non-sequential single ray trace. **Src#** refers to the object number of the desired source. This source must be defined to trace just a single analysis ray. If **Splt?** is non-zero, then splitting is on. If **Pol?** is non-zero then polarization will be used. If splitting is on polarization is automatically selected.
	Scattering is always turned off for this feature, since scattering introduces random paths for the child rays, which is not suitable for optimization. Errors are always considered.
	If multiple NSRA operands trace the same ray are adjacent in the merit function editor, the ray will only be traced once for efficiency.
	Seg# refers to the segment number that contains the data to be returned. Use -1 for the last segment.
	Data refers to the data type for the specified segment.
	Use **Data** values 1-9 for x-coordinate, y-coordinate, z-coordinate, x-cosine, y-cosine, z-cosine, x- normal, y-normal, and z-normal, respectively. These values are relative to the entry port, see 31- 39 below.
	Use **Data** values 10-15 for path-to, intensity, phase of, phase at, index, and starting phase, respectively. Note that the phase at value is not modulo 2*pi, as is the case with ZRD files.
	Use **Data** values 16-17 for the sum of the path, or optical path, respectively, in lens units from the source to the end of the specified segment. These values do not include

	the phase of diffractive surfaces.
	Use **Data** values 21-26 for Ex real, Ex imaginary, Ey real, Ey imaginary, Ez real, and Ez imaginary, respectively. Polarization must be on for real and imaginary data to be returned.
	Use **Data** value 27-29 for the phase of Ex, Ey, and the phase difference between Ex and Ey, respectively. All values are in radians. Polarization must be on for valid data to be returned.
	Use **Data** values 31-39 for the coordinate data defined by **Data** values 1-9 converted to coordinates relative to the global coordinate reference surface.
	Source # refers to the element number to use if the source is an array; see "Modeling arrays of sources" for the numbering scheme.
	For more information on these data items see "Ray database (ZRD) files".
NSRM	See NSRM in "Constraints on non-sequential object data"
NSRW	Non-sequential roadway lighting data.
	Data defines the desired roadway lighting data to be returned. Use 0 for average luminance, 1 for overall uniformity of luminance, 2 for longitudinal uniformity of luminance, 3 for threshold increment,
	4 for surround ratio, 5 for average illuminance, 6 for minimum illuminance, and 7 for uniformity of
	horizontal illuminance. The arguments **# Lanes, Lane Width, Spacing,** and **Offset** are all as
	defined for the roadway lighting analysis. **Arrange** defines the luminaire arrangement along the roadway. Use 0 for single-side, 1 for double-sided, and 2 for staggered. **Surf. Class** defines the
	road surface classification. Use 0 for R1, 1 for R2, 2 for R3, and 3 for R4.
	See "Roadway Lighting" for more information.
	The first NSRW with unique settings must be immediately preceded by an NSTW operand in order to compute data. NSRW operands with the same settings (except **Data**) must be placed on adjacent lines. These subsequent operands will simply return data cached by the first NSRW operand. This operand is intended to be used as part of the NSC Roadway Merit Function Tool.
	See "NSC Roadway Merit Function Tool" for more information.
NSST	Non-sequential single ray trace. This operand traces a single sequential ray through the system to any specified Non-Sequential surface, and returns various data about the ray within the Non- Sequential surface. **Surf** is the surface number of the Non-Sequential surface. **Wave** is the wavelength number. For a definition of **Hx**, **Hy**, **Px**, and

	Py, see "Hx, Hy, Px, and Py". **Data** determines what data NSST will compute and return as follows: 0, 1, 2: The x, y, or z coordinate at the ray-object intercept point. 3, 4, 5: The x, y, or z direction cosines of the ray after reflection/refraction from the surface. 6, 7, 8: The x, y, or z direction cosines of the ray up to the ray-object intercept point. 9, 10, 11: The surface normal at the ray-object intercept point. 12: The face number of the object the ray struck. **Object** specifies which object the desired data is for. In the general case, a ray may strike the same object multiple times. By default, NSST will return the data for the *last* intercept on the specified object. To select a specific intercept, add 1000 to the **Data** value for each time the ray strikes the object. For example, to compute the y coordinate value of the ray on the third time the ray strikes an object, use a **Data** value of 3001. If the ray does not strike the object, or does not strike the object the specified number of times, the operand returns a value of zero, but no warning or error is issued. All coordinates and cosines are in the coordinate system of the Non-Sequential surface. An NSDC, NSDD, NSDE, or NSDP operand with Det# set equal to 0 must be included in the merit function prior to this operand. See also NSTR.
NSTR	Non-sequential trace. **Src#** refers to the object number of the desired source. If **Src#** is zero, all sources will be traced. If **Splt?** is non-zero, then splitting is on. If **Scat?** is non-zero, then scattering is on. If **Pol?** is non-zero then polarization will be used. If splitting is on polarization is automatically selected. If **IgEr?** is non-zero, then errors will be ignored. See "Optimizing with sources and detectors in non-sequential mode" for complete details. An NSDD, NSDE, or NSDP operand with Det# set equal to 0 must be included in the merit function prior to this operand. See also NSST.
NSTW	Non-sequential roadway lighting raytrace. This operand is solely for optimizing roadway lighting applications in conjunction with the NSRW operand and the NSC Roadway Merit Function. If **Splt?** is non-zero, then splitting is on. If **Scat?** is non-zero, then scattering is on. If **Pol?** is non-zero then polarization will be used. If splitting is on polarization is automatically selected. If **IgEr?** is non-zero, then errors will be ignored. **Origin** is the row number of the origin object. **MH** is the mounting height used in the calculation. See "Roadway Lighting" for more information. NSTW should be immediately followed by an NSRW operand to compute data. This

	operand is intended to be used as part of the NSC Roadway Merit Function Tool.
	See "NSC Roadway Merit Function Tool" for more information.
REVR	Reverse Radiance. This operand returns the Total Flux computed by the reverse radiance feature (see "ReverseRadiance™ Analysis"). The **Surf** value is 1 for non-sequential systems, otherwise, it is the surface number of the non-sequential group. **Object** is the number of the Reverse Detector Object, **Pixel #** is 0 to sum all or the individual pixel number, and Data Type is 0 through 3 for total flux or tristimulus X, Y, Z. This operand works independently and does not require the use of other operands for ray tracing or clearing detectors.

Constraints on construction optics for optically fabricated holograms

Operands for constraints on construction optics for optically fabricated holograms

CMFV

NAME	Description
CMFV	Construction merit function value. This operand calls the merit functions defined in either of the two construction systems used to define an optically fabricated hologram. The **Cons#** is either 1 or 2, for the first or second construction system, respectively. The **Op#** is either 0, which will return the entire merit function value from the construction system; or it is an integer which defines the operand row # from which to extract the value from. For example, if **Cons#** is 2 and **Op#** is 7, CMFV will return the value of merit function operand 7 in construction file 2.
	If there are more than one optically fabricated hologram surfaces in the playback system being optimized, the **Cons#** may be incremented by 2 to specify the second surface parameters be used, or by 4 to indicate the third hologram surface construction optics be used, and so on. For example, a **Cons#** of 7 would indicate construction system 1 on the fourth optically fabricated hologram surface present.

Constraints on optical coatings, polarization ray trace data

Operands for constraints on optical coatings, polarization ray trace data

CMGT, CMLT, CMVA, CODA, CEGT, CELT, CEVA, CIGT, CILT, CIVA

NAME	Description
CMGT	Boundary operand that constrains the coating multiplier of the coating layer defined by **Layr** for the surface defined by **Surf** to be greater than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CMLT	Boundary operand that constrains the coating multiplier of the coating layer defined by **Layr** for the surface defined by **Surf** to be less than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CMVA	Boundary operand that constrains the coating multiplier of the coating layer defined by **Layr** for the surface defined by **Surf** to be equal to the target value.
CODA	Coating Data. This feature traces a polarized ray using the system global polarization state (see "Polarization"). Even if the global polarization state is set to "unpolarized", the defined polarization state is used because the data computed by CODA is polarization state specific. The polarized ray may be traced from any field point in object space defined by **Field** to any point in the pupil, to the surface defined by **Surf**. If **Surf** is zero, the ray is traced to the image surface. The wavelength is defined by **Wave** while **Px** and **Py** define the normalized pupil coordinate. The absolute value of **Data** determines what data is returned as follows: 0: The relative transmitted polarized intensity (see 8) 1, 2, 3: The intensity reflectance R, transmittance T, absorption A 4,5: The field amplitude transmittance real, imaginary 6,7: The field amplitude reflectance real, imaginary 8: The relative transmitted unpolarized intensity (see 0) 101, 102: E field out X real, imaginary 103, 104: E field out Y real, imaginary 105, 106: E field out Z real, imaginary 110: The phase difference between Ex and Ey; Pxy 111, 112, 113: The E field phase Px, Py, Pz 121, 122, 123: The major axis, minor axis, and angle in degrees of the polarization ellipse If the data is related to the coating (1-7), and **Data** is negative, the data is for the "S" polarization, otherwise the data is for the "P" polarization. Other coating and polarization data may be added to this list upon request. See "Hx, Hy,

	Px, and Py" .
CEGT	Boundary operand that constrains the coating extinction offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be greater than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CELT	Boundary operand that constrains the coating extinction offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be less than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CEVA	Boundary operand that constrains the coating extinction offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be equal to the target value.
CIGT	Boundary operand that constrains the coating index offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be greater than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CILT	Boundary operand that constrains the coating index offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be less than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CIVA	Boundary operand that constrains the coating index offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be equal to the target value.

Physical Optics Propagation (POP) results

Operands for Physical Optics Propagation (POP) results	
POPD, POPI	
NAME	**Description**
POPD	See "POPD" under Fiber coupling operands.
POPI	Physical Optics Propagation Data. For important details see the section on "Physical

	Optics Propagation".
	To use this operand, first define the settings on the POP analysis feature as desired, then press
	Save on the settings box. The operand will return data based upon the selected settings.
	If **Surf** is zero, then the saved ending surface number will be used; otherwise, the specified surface will be used as the ending surface. If **Wave** is zero, then the saved wavelength number will be used; otherwise, the specified wavelength number will be used. If **Field** is zero, then the saved field number will be used; otherwise, the specified field number will be used.
	Data determines what data the POP feature will compute and return as follows:
	0, 1, 2: The total (Ex + Ey), Ex only, or Ey only Irradiance.
	3, 4, 5, 6: Ex-real, Ex-Imaginary, Ey-real, Ey-imaginary.
	7, 8: Ex, Ey phase in radians. Undefined **Data** values will return 0.
	The **Pix#** refers to the desired pixel of the beam. The pixel number is greater than or equal to zero, and less than nx*ny, where nx and ny are the number of columns and rows, respectively. The pixel number is generally defined as $p = x + y*nx$ where x is the integer row number and y is the integer column number, and $0 <= x < nx$ and $0 <= y < ny$.
	If adjacent POPI operands all have the same **Surf**, **Wave**, and **Field** values, then the POP analysis is done just once and all data returned at one time. Note the POPI operands must be on adjacent rows in the MFE for this efficiency to be implemented.

Best Fit Sphere data

Operands for Best Fit Sphere data

BFSD

NAME	Description
BFSD	Best Fit Sphere (BFS) data. For a complete description, see "Sag Table". The data is computed for the surface defined by **Surf**. The value returned depends upon the value of **Data** as follows:
	0 - BFS curvature in inverse lens units
	1 - BFS radius of curvature in lens units
	2 - Vertex offset from BFS in lens units
	3 - Maximum depth of material to remove in lens units
	4 - Total volume of material to use in cubic lens units
	5 - The maximum slope (dz/dr) difference between the surface and BFS (dimensionless)
	6 - RMS of the depth of material to remove in lens units
	7 - RMS of the slope (dz/dr) difference between the surface and BFS (dimensionless)
	The **MinR** and **MaxR** are the minimum and maximum radial coordinate over which the

	BFS data is computed. If both values are zero, the minimum radial coordinate used is zero and the maximum radial coordinate used is the semi-diameter of the surface.

Tolerance sensitivity data

Operands for tolerance sensitivity data	
TOLR	
NAME	**Description**
TOLR	Tolerance data. **Data** is 0 for RSS estimated change in performance, 1 for nominal performance, and 2 for estimated performance (nominal plus estimated change). **File** is the integer number corresponding to the tolerance settings file to use. **Config#** is -2 for the configuration defined by the last CONF operand, -1 for all configurations, 0 to use the configuration set in the tolerance settings file and 1 or greater for a specific configuration number. For details, see "Optimizing tolerance sensitivity" .

Thermal Coefficient of Expansion data

Operands for Thermal Coefficient of Expansion data	
TCGT, TCLT, TCVA	
NAME	**Description**
TCGT	Thermal Coefficient of expansion greater than. This boundary operand constrains the TCE of the surface defined by **Surf** to be greater than the specified target value.
TCLT	Thermal Coefficient of expansion less than. This boundary operand constrains the TCE of the surface defined by **Surf** to be less than the specified target value.
TCVA	Thermal Coefficient of expansion value. This boundary operand constrains the TCE of the surface defined by **Surf** to be equal to the specified target value. For glass surfaces, see "GTCE".

Obsolete Operands

Obsolete Operands	
NAME	**Description**

PnGT	This operand is obsolete, use PMGT instead.
PnLT	This operand is obsolete, use PMLT instead.
PnVA	This operand is obsolete, use PMVA instead.

Optimization Operands (Alphabetically)

This section provides the a detailed description of each operand, listed alphabetically in a single table. Names shown in **bold** define data values used to define the operand. These same names will appear in the editor column headings.

NAME	Description
ABCD	The ABCD values used by the grid distortion feature to compute generalized distortion. See "Grid Distortion". The reference field number is defined by **Ref Fld**. The wavelength number is defined by **Wave**. **Data** is 0 for A, 1 for B, 2 for C, and 3 for D. See also "DISA".
ABGT	Absolute value of operand greater than. This is used to make the absolute value of the operand defined by **Op#** greater than the target value.
ABLT	Absolute value of operand less than. This is used to make the absolute value of the operand defined by **Op#** less than the target value.
ABSO	Absolute value of the operand defined by **Op#**.
ACOS	Arc cosine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
AMAG	Angular magnification. This is the ratio of the image to object space paraxial chief ray angles at the wavelength defined by **Wave**. Not valid for non-paraxial systems.
ANAC	Angular aberration radial direction measured in image space with respect to the centroid at the wavelength defined by **Wave**. This quantity is defined as: $$\varepsilon = \sqrt{(l - l_c)^2 + (m - m_c)^2}$$ where l and m are the x and y direction cosines of the ray and the c subscript indicates the centroid. See "Hx, Hy, Px, and Py".

ANAR	Angular aberration radius measured in image space at the wavelength defined by **Wave** with respect to the primary wavelength chief ray. This quantity is defined as: $$\varepsilon = \sqrt{(l - l_c)^2 + (m - m_c)^2}$$ where l and m are the x and y direction cosines of the ray and the c subscript indicates the chief ray. See "Hx, Hy, Px, and Py" .
ANAX	Angular aberration x direction measured in image space at the wavelength defined by **Wave** with respect to the primary wavelength chief ray. This quantity is defined as: $$\varepsilon = l - l_c$$ where l is the x direction cosine of the ray and the c subscript indicates the chief ray. See "Hx, Hy, Px, and Py".
ANAY	Angular aberration y direction measured in image space at the wavelength defined by **Wave** with respect to the primary wavelength chief ray. This quantity is defined as: $$\varepsilon = m - m_c$$ where m is the y direction cosine of the ray and the c subscript indicates the chief ray. See "Hx, Hy, Px, and Py".
ANCX	Angular aberration x direction measured in image space at the wavelength defined by **Wave** with respect to the centroid. This quantity is defined as: $$\varepsilon = l - l_c$$ where l is the x direction cosine of the ray and the c subscript indicates the centroid. ANCX has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py".
ANCY	Angular aberration y direction measured in image space at the wavelength defined by **Wave** with respect to the centroid. This quantity is defined as: $$\varepsilon = m - m_c$$ where m is the y direction cosine of the ray and the c subscript indicates the centroid. ANCY has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py".
ASIN	Arcsine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.

ASTI	Astigmatism in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order astigmatism calculated from the Seidel coefficients, and is not valid for non-paraxial systems.
ATAN	Arctangent of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
AXCL	Axial color, measured in lens units for focal systems and diopters for afocal systems. This is the image separation between the two wavelengths defined by **Wave1** and **Wave2**. If **Zone** is zero, paraxial rays are used to determine the paraxial image locations. If **Zone** is greater than 0.0 and less than or equal to 1.0, real marginal rays are used to determine the image locations. In this case, **Zone** corresponds to the Py coordinate of the real marginal ray. See "Hx, Hy, Px, and Py".
BFSD	Best Fit Sphere (BFS) data. For a complete description, see "Sag Table". The data is computed for the surface defined by **Surf**. The value returned depends upon the value of **Data** as follows: 0 - BFS curvature in inverse lens units 1 - BFS radius of curvature in lens units 2 - Vertex offset from BFS in lens units 3 - Maximum depth of material to remove in lens units 4 - Total volume of material to use in cubic lens units 5 - The maximum slope (dz/dr) difference between the surface and BFS (dimensionless) 6 - RMS of the depth of material to remove in lens units 7 - RMS of the slope (dz/dr) difference between the surface and BFS (dimensionless) The **MinR** and **MaxR** are the minimum and maximum radial coordinate over which the BFS data is computed. If both values are zero, the minimum radial coordinate used is zero and the maximum radial coordinate used is the semi-diameter of the surface.
BIOC	Biocular Convergence. Returns the convergence between two eye configurations in milliradians. The left and right eye configurations are specified using the **Left** and **Right** values. The other parameters are: **Wave**: The wavelength number to use. **UseCos**: If 0 field units are degrees, otherwise field is in direction cosine units.

	Xang/**Yang**: The X and Y angle or cosines at which to compute the convergence. If the chief rays from both configurations at the specified angles do not pass through to the image without vignetting, an error is reported. See "Divergence/Convergence" for more information and important assumptions.
BIOD	Biocular Dipvergence. Returns the dipvergence between two eye configurations in milliradians. See BIOC above for details.
BIPF	Unused.
BLNK	Does nothing. Used for separating portions of the operand list. A comment line may optionally be typed in the space to the right of the operand name; this comment will be displayed in the editor as well as in the merit function listing.
BLTH	Blank thickness. Computes the minimum thickness of the glass blank required to create the volume following the surface defined by **Surf**. The sag of the surface and the one following are computed at 200 radial points along axes defined by the **Code** value. Use **Code** = 0 for +Y axis, 1 for +X axis, 2 for -Y axis, 3 for -X axis, and 4 for all four axes.
BSER	Boresight error. Boresight error is defined as the radial chief ray coordinate traced for the on axis field and wavelength defined by **Wave** divided by the effective focal length. This definition yields a measure of the angular deviation of the image.
CEGT	Boundary operand that constrains the coating extinction offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be greater than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CELT	Boundary operand that constrains the coating extinction offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be less than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CEHX	Huygens PSF centroid X position. This operand uses the Huygens PSF to determine the x coordinate of the centroid for any field point. The centroid accounts for all apodization and apertures, and optionally polarization. The parameters for this operand are: **Wave**: The wavelength number to use (0 for polychromatic, otherwise use the monochromatic wavelength number). **Field:** The integer field number to use. **Pol?:** Set to 0 to ignore polarization and 1 to consider it.

	Pupil Samp: Use 1 for 32x32, 2 for 64x64, etc. **Image Samp**: Use 1 for 32x32, 2 for 64x64, etc. **All Conf?:** Set to 0 to use the current configuration (defined by the last CONF operand preceding this operand), and 1 to sum over all configurations. See "Huygens PSF" for a full discussion of this option. This feature always uses the default image delta. The evaluation surface is always the image surface (see IMSF). When CEHX is followed by a CEHY with identical settings, both are computed with the same ray set to save time. See also CENX, CENY, CNPX, CNPY, CNAX, and CNAY.
CEHY	Huygens PSF centroid Y position. See CEHX.
CENX	Centroid X position. This operand uses a grid of rays to determine the x coordinate of the centroid of all rays from a single field point. The centroid accounts for apodization and apertures, and optionally polarization. The parameters for this operand are: **Surf**: The surface number to use (use 0 for the image surface). **Wave**: The wavelength number to use (0 for polychromatic, otherwise use the monochromatic wavelength number). **Field:** The integer field number to use. **Pol?:** Set to 0 to ignore polarization and 1 to consider it. **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. When CENX is followed by a CENY with identical settings, both are computed with the same ray set to save time. See also CEHX, CEHY, CNPX, CNPY, CNAX, and CNAY.
CENY	Centroid Y position. See CENX.
CEVA	Boundary operand that constrains the coating extinction offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be equal to the target value.
CIGT	Boundary operand that constrains the coating index offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be greater than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CILT	Boundary operand that constrains the coating index offset of the coating layer defined

	by **Layr** for the surface defined by **Surf** to be less than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CIVA	Boundary operand that constrains the coating index offset of the coating layer defined by **Layr** for the surface defined by **Surf** to be equal to the target value.
CMFV	Construction merit function value. This operand calls the merit functions defined in either of the two construction systems used to define an optically fabricated hologram. The **Cons#** is either 1 or 2, for the first or second construction system, respectively. The **Op#** is either 0, which will return the entire merit function value from the construction system; or it is an integer which defines the operand row # from which to extract the value from. For example, if **Cons#** is 2 and **Op#** is 7, CMFV will return the value of merit function operand 7 in construction file 2. If there are more than one optically fabricated hologram surfaces in the playback system being optimized, the **Cons#** may be incremented by 2 to specify the second surface parameters be used, or by 4 to indicate the third hologram surface construction optics be used, and so on. For example, a **Cons#** of 7 would indicate construction system 1 on the fourth optically fabricated hologram surface present.
CMGT	Boundary operand that constrains the coating multiplier of the coating layer defined by **Layr** for the surface defined by **Surf** to be greater than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CMLT	Boundary operand that constrains the coating multiplier of the coating layer defined by **Layr** for the surface defined by **Surf** to be less than the target value. Use **Surf** = 0 for all surfaces, **Layr** = 0 for all layers.
CMVA	Boundary operand that constrains the coating multiplier of the coating layer defined by **Layr** for the surface defined by **Surf** to be equal to the target value.
CNAX	Centroid angular x direction. This operand computes the X angle in radians relative to the local Z axis of the centroid of rays from any field point. The centroid accounts for apodization and apertures, and optionally polarization. **Surf**: The surface number to use (use 0 for the image surface). **Wave**: The wavelength number to use (0 for polychromatic, otherwise use the monochromatic wavelength number). **Hx/Hy:** The normalized field coordinates to use. See "Hx, Hy, Px, and Py". **Pol?:** Set to 0 to ignore polarization and 1 to consider it. **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. When CNAX is followed by a CNAY with identical settings, both are computed with the

	same ray set to save time. See also CNAY, CNPX, CNPY, CENX, CENY, CEHX, CEHY.
CNAY	Centroid angular y direction. See CNAX.
CNPX	Similar to CNAX, but computes the centroid position rather than angle.
CNPY	Similar to CNAY, but computes the centroid position rather than angle.
CODA	Coating Data. This feature traces a polarized ray using the system global polarization state (see "Polarization"). Even if the global polarization state is set to "unpolarized", the defined polarization state is used because the data computed by CODA is polarization state specific. The polarized ray may be traced from any field point in object space defined by **Field** to any point in the pupil, to the surface defined by **Surf**. If **Surf** is zero, the ray is traced to the image surface. The wavelength is defined by **Wave** while **Px** and **Py** define the normalized pupil coordinate. The absolute value of **Data** determines what data is returned as follows: 0: The relative transmitted polarized intensity (see 8) 1, 2, 3: The intensity reflectance R, transmittance T, absorption A 4,5: The field amplitude transmittance real, imaginary 6,7: The field amplitude reflectance real, imaginary 8: The relative transmitted unpolarized intensity (see 0) 101, 102: E field out X real, imaginary 103, 104: E field out Y real, imaginary 105, 106: E field out Z real, imaginary 110: The phase difference between Ex and Ey; Pxy 111, 112, 113: The E field phase Px, Py, Pz 121, 122, 123: The major axis, minor axis, and angle in degrees of the polarization ellipse If the data is related to the coating (1-7), and **Data** is negative, the data is for the "S" polarization, otherwise the data is for the "P" polarization. Other coating and polarization data may be added to this list upon request. See "Hx, Hy, Px, and Py" .
COGT	Boundary operand that constrains the conic of the surface defined by **Surf** to be

	greater than the specified target value.
COLT	Boundary operand that constrains the conic of the surface defined by **Surf** to be less than the specified target value.
COMA	Coma in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order coma calculated from the Seidel coefficients, and is not valid for non-paraxial systems.
CONF	Configuration. This operand is used to change the configuration number to the configuration defined by **Cfg#** during merit function evaluation. This permits optimization across multiple configurations with a single merit function. This operand does not use the target or weight columns.
CONS	Constant value. This is used to enter in constant values for use in other operand computations. The value will be identical to the target value.
COSA	Unused.
COSI	Cosine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
COVA	Conic value. Returns the conic constant of the surface defined by **Surf**.
CTGT	Center thickness greater than. This boundary operand constrains the center thickness of the surface defined by **Surf** to be greater than the specified target value. See also MNCT.
CTLT	Center thickness less than. This boundary operand constrains the center thickness of the surface defined by **Surf** to be less than the specified target value. See also MXCT.
CTVA	Center thickness value. Constrains the center thickness of the surface defined by **Surf** to be equal to the specified target value.
CVGT	Curvature greater than. This boundary operand constrains the curvature of the surface defined by **Surf** to be greater than the target value.
CVIG	Clears the vignetting factors. This operand clears the current vignetting factors for the remainder of the operands in the current configuration. See also "SVIG".
CVLT	Curvature less than. This boundary operand constrains the curvature of the surface

	defined by **Surf** to be less than the target value.
CVOL	Cylinder volume. This operand computes the volume in cubic lens units of the smallest cylinder that will contain the range of surfaces defined by **Surf1** and **Surf2**. Only the vertex positions and semi- diameters are used in the calculation, not the sag. The range of surfaces should not include any coordinate breaks.
CVVA	Curvature value. This operand constrains the curvature of the surface defined by **Surf** to be equal to the specified target value.
DENC	Diffraction Encircled Energy (distance). This operand computes the distance to the fraction of diffraction encircled, ensquared, x only, or y only (enslitted) energy defined by **Frac**. For focal mode, the units are micrometers. For afocal mode, the units are afocal mode units. The other parameters are: **Samp:** The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Type**: 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. **Refp**: The reference point/algorithm to use. For FFT encircled energy, use **Refp** = 0 for chief ray, 1 for centroid, and 2 for vertex. For Huygens encircled energy, use **Refp** = 3 for chief ray, 4 for centroid, and 5 for vertex. When using the Huygens method, **I Samp** refers to the Huygens image sampling (1 for 32 x 32, 2 for 64 x 64, etc.) and **I Delta** refers to the Huygens image delta. For a detailed description of the Huygens image sampling and image delta, see "Encircled Energy". If the sampling is too low, the radius returned is 1e+10. See also DENF, GENC and XENC.
DENF	Diffraction Encircled Energy (fraction). This operand computes the fraction of diffraction encircled, ensquared, x only, or y only (enslitted) energy at a given distance from the reference point defined by **Dist**. For focal mode, the distance units are micrometers. For afocal mode, the distance units are afocal mode units. The options and settings are identical to DENC, except **Dist**, which here is used as the distance at which the fraction of energy is desired. See also DENC, GENC, GENF, and XENC. If the **Dist** value defined is beyond the point where the encircled energy is very close to 100%, the fraction returned is 1e+10; this is done for optimization efficiency.
DIFF	Difference of two operands (**Op#1** - **Op#2**).

DIMX	Distortion maximum. This is similar to DIST, except it specifies only an upper bound for the absolute value of the distortion. **Field** can be zero, which specifies that the maximum field coordinate be used, or any valid field number. Note the maximum distortion does not always occur at the maximum field coordinate. The distortion is calculated at the wavelength defined by **Wave**.
	If **Absolute** is 0, the value returned is in units of percentage. If **Absolute** is 1, the distortion is given as an absolute length rather than a percentage.
	This operand may not be valid for non-rotationally symmetric systems.
DISA	Distortion, ABCD. This operand computes the Radial, X, or Y direction distortion relative to the reference field (**Ref Fld**), for the chief ray at the wavelength defined by **Wave. Data** is 0 for radial distortion, 1 for X direction distortion, and 2 for Y direction distortion. The distortion is computed for the chief ray at the field point defined by **Field**. The **A**, **B**, **C**, **D** values are user defined. The distortion is computed in the same manner as the grid distortion feature (see "Grid Distortion"). The key difference between this operand and DISG is that the ABCD values are user defined. See also "ABCD" and "DISG".
DISC	Distortion, calibrated. This operand computes the calibrated f-theta distortion across the y-field of view at the wavelength defined by **Wave**, and returns the absolute value of the maximum deviation from linearity of the f-theta condition.
	If **Absolute** is 0, the value returned is in units of percentage. If **Absolute** is 1, the distortion is given as an absolute length (for focal systems) or an absolute deviation in cosine space (for afocal systems) rather than a percentage. This operand is useful for designing f-theta lenses.
DISG	Generalized distortion, either in percent or as an absolute distance. This operand computes the distortion for any ray in the pupil, from anywhere in the field, at the wavelength defined by **Wave**, using the field point defined by **Field** as a reference. The method used and assumptions made are the same as for the grid distortion plot, see "Grid Distortion". DISG cannot be calculated if the field units are angles and the maximum angle equals or exceeds 90 degrees. DISG assumes the predicted magnification is not symmetric.
	If the field is defined in terms of angles, the normalized field coordinates **Hx** and **Hy** are defined differently than for other Zemax features. In this case only, **Hx** and **Hy** correspond to divisions on a uniform projected grid in object space, centered on the reference field coordinate. These normalized field coordinates are defined as
	$$H = \frac{\tan(\theta)}{\tan(\theta_M)}$$
	where θ is the angle relative to the reference field chief ray and θ_M is the maximum field angle (see "Maximum field"). These definitions are needed to support the reference field feature.

	If **Wave** is a positive number, DISG returns the distortion as a percentage. If **Wave** is a negative number, the absolute value of **Wave** is used to define the wavelength and the returned value is in units of absolute length rather than percentage.
	As with all distortion concepts, the best way to avoid confusion and misleading results is to use finite object distances and object heights to define fields rather than field angles.
	See "Hx, Hy, Px, and Py".
DIST	Distortion in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. This is the third order distortion calculated from the Seidel coefficients (see "Seidel Coefficients"), and is not valid for non-paraxial systems.
	If **Surf** is zero, the distortion is given in percent instead (see "Field Curvature/Distortion" for a detailed definition).
	If **Absolute** is set to 1, and the surface number is zero, the distortion is given as an absolute length rather than a percentage.
	See also DISG.
DIVB	Divides the value of any prior operand defined by **Op#** by any factor defined by **Factor**.
DIVI	Division of first by second operand (**Op#1** / **Op#2**). See also "RECI".
DLTN	Delta N. Computes the difference between the maximum and minimum index of refraction on axis for a gradient index surface defined by **Surf**. The wavelength used is defined by **Wave**. The minimum and maximum z coordinates used are computed by accounting for the sag of both ends of the surface. See the section "Using gradient index operands".
DMFS	Default merit function start. This operand is a flag to indicate where the default merit function should be appended, if one is subsequently created. The row number after this operand will appear as the default "Start At" row on the default merit function dialog box.
DMGT	Diameter greater than. This boundary operand constrains the diameter of the surface defined by **Surf** to be greater than the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.
DMLT	Diameter less than. This boundary operand constrains the diameter of the surface defined by **Surf** to be less than the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.
DMVA	Diameter value. This operand constrains the diameter of the surface defined by **Surf** to be equal to the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.

DXDX	Derivative of transverse x-aberration with respect to x-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py" .
DXDY	Derivative of transverse x-aberration with respect to y-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
DYDX	Derivative of transverse y-aberration with respect to x-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
DYDY	Derivative of transverse y-aberration with respect to y-pupil coordinate. This is the slope of the ray fan plot at the specified pupil coordinate at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
EFFL	Effective focal length in lens units. The wavelength used is defined by **Wave**. This is paraxial, and may not be accurate for non-paraxial systems.
EFLX	Effective focal length in the local x plane of the range of surfaces defined by **Surf1** and **Surf2** at the primary wavelength.
EFLY	Effective focal length in the local y plane of the range of surfaces defined by **Surf1** and **Surf2** at the primary wavelength.
EFNO	Effective F/#. This operand computes the Effective F/# of the field point defined by **Field**. For a discussion of Effective F/# see "Effective F/#". The other parameters are: **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. **Wave**: The wavelength number to use. **Pol?**: Set to 0 to ignore polarization and 1 to consider it. See also RELI.
ENDX	End execution. Terminates the computation of the merit function. All remaining operands are ignored.
ENPP	Entrance pupil position in lens units, with respect to the first surface. This is the paraxial pupil position, valid only for centered systems.
EPDI	Entrance pupil diameter in lens units.

ERFP	Edge Response Function Position. This operand computes the x or y position of the point at which the edge response function reaches a certain relative value. For details on the edge response function calculation, see "Geometric Line/Edge Spread". The **Sampling** value is 1 for 32x32, 2 for 64x64, etc. **Wave** is the wavelength number or 0 for polychromatic. **Field** is the field number. **Type** determines that data to be returned. If **Type** is 0 or 1, the x position (for edges parallel to the local y axis) or y position (for edges parallel to the local x axis) relative to the chief ray in lens units is returned. If **Wave** is zero, the primary wavelength chief ray is the reference point. If **Type** is 2 or 3, the x or y position in lens units relative to the surface vertex is returned. **Fraction** is the relative value of the edge response curve, and must be between 0.01 and 0.99. **Max Radius** is the maximum radial size of the integration window in micrometers. If **Max Radius** is zero a default value is used, and this is the recommended setting in most cases. Note that the edge response function is normally defined with the "bright" side of the edge being on the + side of the integration coordinate, which means the edge response goes to 1 as the coordinate becomes more positive. To compute results for the reversed edge orientation, with the bright side on the negative side of the coordinate, use the value (1-fraction) instead of (fraction). For example, to get the 80% response coordinate for a reversed edge, use **Fraction** = 0.20. If afocal mode is used, all returned values are in afocal analysis units.
EQUA	Equal operand. This operand constrains all operands within the range of operands defined by **Op#1** and **Op#2** to have the same value within the tolerance specified by the target. The value of this operand is computed by finding the average of the range of values, and then summing the absolute value of the errors between each operand and the average if the error exceeds the target value. See SUMM and OSUM.
ETGT	Edge thickness greater than. This boundary operand constrains the edge thickness of the surface defined by **Surf** to be greater than the specified target value. The edge thickness is calculated along the +y axis if **Code** is zero, the +x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. See also MNET.
ETLT	Edge thickness less than. This boundary operand constrains the edge thickness of the surface defined by **Surf** to be less than the specified target value. The edge thickness is calculated along the +y axis if **Code** is zero, the +x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. See also MXET.
ETVA	Edge thickness value. Constrains the edge thickness of the surface defined by **Surf** to be equal to the specified target value. The edge thickness is calculated along the +y axis if **Code** is zero, the +x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. See also MNET.

EXPD	Exit pupil diameter in lens units. This is the paraxial pupil diameter, valid only for centered systems.
EXPP	Exit pupil position in lens units, with respect to the image surface. This is the paraxial pupil position, valid only for centered systems.
FCGS	Generalized field curvature, sagittal. The field curvature value for any field point, at the wavelength defined by **Wave**. The value is generalized to return reasonable results even for non-rotationally symmetric systems; see "Field Curvature/Distortion. See "Hx, Hy, Px, and Py".
FCGT	Generalized field curvature, tangential; see FCGS.
FCUR	Field curvature in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order field curvature calculated from the Seidel coefficients, and is not valid for non-paraxial systems.
FDMO	Field Data Modify. This operand allows temporary modification of field position data. The operand allows new field coordinates and vignetting factors (except for vignetting angle, which cannot be modified by this operand) for any field coordinate. All subsequent operands will use the modified field data. The original field data is restored when a FDRE operand is reached with the same field number argument, or a CONF operand is reached (regardless of the configuration number referenced by the CONF operand), or after the last operand in the merit function is reached. **Field**: The field number to modify. **Hx, Hy**: The normalized field coordinates for the new field position. **VDX, VDY, VCX, VCY**: The vignetting factor data for the new field position. See "Vignetting factors".
FDRE	Field Data Restore. See FDMO. **Field**: The field number to restore.
FICL	Fiber coupling efficiency for single mode fibers. The calculated value is the total coupled energy efficiency, relative to unity. The parameters of this operand are: **Samp**: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. **Wave**: The wavelength number to use. **Field**: The integer field position number. **IgSrc**: If this parameter is zero, then the object source fiber is considered; otherwise,

	the object source fiber is ignored. **Sna**: The source fiber NA. **Rna**: The receiver fiber NA. **Data**: This controls what data is returned as well as the algorithm to use. If **Data** is 0, the fast algorithm is used and the power coupling is returned. If **Data** is 1, the Huygens algorithm is used and the power coupling is returned. If **Data** is 2 or 4, the fast algorithm is used and the real or imaginary part of the amplitude coupling is returned, respectively. If **Data** is 3 or 5, the Huygens algorithm is used and the real or imaginary part of the amplitude coupling is returned, respectively. **Pol?**: Set to 0 to ignore polarization and 1 to consider it. See "Fiber Coupling Efficiency" for details. See also FICP.
FICP	Fiber coupling as computed using the Physical Optics Propagation (POP) algorithm, using whatever the current default settings are for the POP feature. **Data**: This controls what data is returned. If **Data** is 0 the total power coupling is returned. If **Data** is 1, the amplitude coupling for the Ex field is returned. If **Data** is 2 or 3, the real or imaginary part of the amplitude coupling for the Ex field is returned, respectively. If **Data** is 4, the amplitude coupling for the Ey field is returned. If **Data** is 5 or 6, the real or imaginary part of the amplitude coupling for the Ey field is returned, respectively. If the beam is unpolarized the Ey values wil be zero. To use this operand, first define the settings on the POP analysis feature as desired, then press Save on the settings box. The operand FICP will return the same efficiency as computed by the POP feature. This operand is redundant with the more general POPD. See "Computing Fiber Coupling". See also FICL.
FOUC	Foucault analysis. This operand returns the RMS difference between the computed and reference shadowgram as computed by the Foucault analysis feature, using whatever the current default settings are. To use this operand, first define the settings on the Foucault analysis feature as desired, then press Save on the settings box. The operand FOUC will return the RMS difference between the computed and reference shadowgrams, regardless of how the "data" option is set in the saved configuration file. Using this operand, the optical system wavefront aberrations may be optimized to produce the reference shadowgram.
FREZ	Freeform Z object constraints. This operand computes various values from the shape of a Freeform Z object described in "Freeform Z". **Surf** is the surface number of the NSC group, use 1 for NSC program mode. **Object** is the object number, which must be a Freeform Z object for this

	operand to compute any data. The **Data** value determines what value is computed by the operand
	as follows:
	1: Maximum z value. Nnote minimum z value is always zero.
	2: Maximum z increment. This is the biggest difference between adjacent z control points.
	3: Minimum z increment. This is the smallest difference between adjacent z control points.
	4: Minimum y value. This value is for the entire length of the solid, not just at control points.
	5: Maximum y value. This value is for the entire length of the solid, not just at control points.
	6: Maximum y increment. This is the biggest difference between adjacent y control points.
	7: Minimum y increment. This is the smallest difference between adjacent y control points.
	8: Volume of the object in lens units cubed.
	9: Monotonic deviation. This is the largest change in adjacent y values whose sign deviates from the sign of the first y change going from 0 to positive z coordinates.
	10: Minimum slope.
	11: Maximum slope.
	Once the **Data** has been computed, the **Mode** value determines how the operand behaves. If **Mode** is 1, the FREZ operand returns the data. If **Mode** is 2, the operand returns the data value only if the data value is less than the target value, otherwise the target value is returned. This mode is used to enforce a "greater than" boundary on the data. If **Mode** is 3, the operand returns the data value only if the data value is greater than the target value, otherwise the target value is returned. This mode is used to enforce a "less than" boundary on the data.
FTGT	Full thickness greater than. This boundary operand constrains the full thickness of surface **Surf** to be greater than the specified target value. The full thickness is computed at 200 points between the vertex and edge along the +y radial direction, including the sag of the surface and the sag of the next surface. The operand is useful for constraining surfaces which do not have their minimum or maximum thickness at the center or edge, but at some intermediate zone. See FTLT.
FTLT	Full thickness less than. See FTGT.

GBPD	Gaussian beam (paraxial) divergence in the optical space following the surface defined by **Surf** at the wavelength defined by **Wave**. The other parameters are: **UseX**: If this parameter is non-zero, then the computation is for the x-direction beam, otherwise, it is for the y-direction. **W0**: The input beam waist size in lens units. **S1toW**: The distance from Surface 1 to the waist location in lens units. See the Gaussian beam feature for details. **M2 Factor**: The M Squared value for the beam. See the Gaussian beam feature for details.
GBPP	Gaussian beam (paraxial) position, which is the distance from the waist to the surface, in the optical space following the specified surface. See GBPD.
GBPR	Gaussian beam (paraxial) radius of curvature in the optical space following the specified surface. See GBPD.
GBPS	Gaussian beam (paraxial) size in the optical space following the specified surface. See GBPD.
GBPW	Gaussian beam (paraxial) waist in the optical space following the specified surface. See GBPD.
GBPZ	Gaussian beam (paraxial) Rayleigh range in the optical space following the specified surface. See GBPD.
GBSD	Gaussian beam (skew) divergence in the optical space following the specified surface. The input beam is aligned along the chief ray of the field defined by **Field**. The other parameters are: **In#**: The surface number to start the propagation. **Out#**: The surface at which to compute the Skew Gaussian Beam data. **Wave**: The wavelength number to use. **StoW**: The starting surface to waist distance in lens units. **W0**: The input beam waist size in lens units. If **W0** is positive, then the computation is for the y- direction beam, otherwise, it is for the x-direction. For details on the Skew Gaussian Beam feature, see "Skew Gaussian Beam".
GBSP	Gaussian beam (skew) position, which is the distance from the waist to the surface, in

	the optical space following the specified surface. See GBSD.
GBSR	Gaussian beam (skew) radius in the optical space following the specified surface. See GBSD.
GBSS	Gaussian beam (skew) size in the optical space following the specified surface. See GBSD.
GBSW	Gaussian beam (skew) waist in the optical space following the specified surface. See GBSD.
GCOS	Glass cost. This operand returns the relative cost factor as listed in the glass catalog for the glass on the surface defined by **Surf**.
GENC	Geometric Encircled Energy (distance). This operand computes the distance to the fraction of geometric encircled, ensquared, x only, or y only (enslitted) energy defined by **Frac**. For focal mode, the units are micrometers. For afocal mode, the units are afocal mode units. The other parameters are: **Samp**: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Type**: 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. **Refp**: The reference point to use. Use 0 for chief ray, 1 for centroid, 2 for vertex, and 3 for middle of the spot. **No Diff Lim**: If 0, the results are scaled by the diffraction limit, otherwise, no accounting of diffraction is done. See also GENF, DENC, DENF, and XENC.
GENF	Geometric Encircled Energy (fraction). This operand computes the fraction of geometric encircled, ensquared, x only, or y only (enslitted) energy at a given distance from the reference point defined by **Dist**. The options and settings are identical to GENC, except **Dist**, which here is used as the distance at which the fraction of energy is desired. See also GENC, DENC, DENF, and XENC.
GLCA	Global x-direction orientation vector component of the surface defined by **Surf**.
GLCB	Global y-direction orientation vector component of the surface defined by **Surf**.

GLCC	Global z-direction orientation vector component of the surface defined by **Surf**.
GLCR	Global Coordinate Rotation Matrix component at the surface defined by **Surf**. The 3 x 3 R matrix has 9 components. If **Data** is 1, GLCR returns R[1][1], if **Data** is 2, GLCR returns R[1][2], etc... through **Data** = 9 returning R[3][3].
GLCX	Global vertex x-coordinate of the surface defined by **Surf**.
GLCY	Global vertex y-coordinate of the surface defined by **Surf**.
GLCZ	Global vertex z-coordinate of the surface defined by **Surf**.
GMTA	Geometric MTF average of sagittal and tangential response. The parameters are: **Samp**: Higher sampling yields higher accuracy at the expense of computation time. To confirm the computation has acceptable accuracy, start at 1, and increment the sampling until the results change by less than the desired accuracy. Note that extreme precision is not required for good optimization results; three significant figures is usually adequate. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Freq**: The spatial frequency in MTF units (see "MTF Units"). **!Scl**: If zero, then the diffraction limit will be used to scale the results (recommended) otherwise, no scaling is done. If **Grid** is zero, a fast, sparse sampling integration method is used to compute the MTF. The fast Geometric MTF algorithm is only accurate for systems with circular or elliptical pupils with modest or no apodization. For systems that violate this assumption, set **Grid** equal to 1. The fast sampling method used by GMTA, GMTS, and GMTT is not directly related to the Geometric MTF Analysis feature. Because only a single spatial frequency is required, the method of computation used by the MTF operands is different, and generally much faster, than the algorithm used by the analysis feature. To select the alternate grid-based algorithm used by the Geometric MTF analysis feature, set **Grid** equal to 1. The grid-based algorithm is usually slower than the default algorithm if the MTF is reasonably good (greater than 5%), but the grid algorithm converges faster if the aberrations are very large and the resulting MTF is very low. If both the tangential and sagittal MTF are needed; place the GMTT and GMTS operands on adjacent lines and they will be computed simultaneously. The Geometric MTF, though approximate, can usually be computed more quickly than the Diffraction MTF, and is therefore useful for optimization. See "Performing an optimization".
GMTS	Geometric MTF sagittal response. See GMTA.

GMTT	Geometric MTF tangential response. See GMTA.
GOTO	Skips all operands between the GOTO operand line and the operand number defined by **Op#**. Execution of the merit function will start again at the **Op#** line.
GPIM	Ghost pupil image. GPIM controls the location of ghost pupils (and optionally ghost images) relative to the image surface. Double-bounce ghosts form images of the pupil, and if these images are formed near the focal plane they will contaminate the image with unwanted light. This is the cause of the familiar "sun flare" images of the pupil seen through camera lenses pointed near the sun.
	The operand computes any one specific or all possible ghost pupil image locations and returns one over the absolute value of the distance from the image surface to the closest pupil ghost. The operand is defined in this manner so it can be targeted to zero and weighted and optimized to reduce ghost pupil affects. If the **Surf1** and **Surf2** parameters are set to specific surface numbers, that specific ghost path is computed, if either or both of the **Surf1** and **Surf2** values are -1, then all possible surface combinations are considered. For example, if **Surf1** is 12 and **Surf2** is -1, then all double bounces that first bounce off surface 12 and then 11, 10, 9, etc. are considered. If both numbers are negative, all possible ghosts are considered.
	This same operand also can be used for detecting and controlling image ghosts (which are distinct from pupil ghosts) by changing **Mode** from 0 to 1, or to control ghost pupil magnification (the ratio of the ghost exit pupil diameter to entrance pupil diameter), by setting **Mode** to 2.
	The **WFB** and **WSB** columns will list the worst combination found for reference and possible further analysis. Only surfaces with index changes are considered as possible ghost generators. First bounces off mirrors are ignored.
	See also GPRT, GPRX, GPRY, GPSX, and GPSY.
GPRT	Ghost ray transmission. This operand computes the unpolarized transmission of any ray from object to image surface for a double-bounce ghost path. The **Surf1** and **Surf2** values define the first and second surface numbers to define the ghost path. Note the first bounce surface number must be larger than the second bounce surface number. The primary wavelength is always used.
	GPRT makes no changes to the field or aperture specifications, and therefore may return meaningless results if the field or aperture data are defined in a way that changes the ghost path. No warning is issued in these cases. For example, using GPRT while using image height for field points or using image space F/# for the system aperture type will likely return useless results. The recommended approach is to use field angles or object heights, use entrance pupil diameter, and place the stop prior to the first bounce surface. To study the double bounce system in detail see "Ghost Focus Generator".
	See "Hx, Hy, Px, and Py.

GPRX	Ghost ray real x coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the x coordinate of the real ray on the image surface.
GPRY	Ghost ray real y coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the y coordinate of the real ray on the image surface.
GPSX	Ghost ray paraxial x coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the x coordinate of the paraxial ray on the image surface.
GPSY	Ghost ray paraxial y coordinate. This operand is similar to GPRT, see that discussion for assumptions and limitations. The returned value is the y coordinate of the paraxial ray on the image surface.
GRMN	Gradient index minimum index. This boundary operand sets the minimum allowable index of refraction value for the gradient index surface defined by **Surf** at the wavelength defined by **Wave**. The index is checked at six places: the front vertex, front +y top, front +x side, rear vertex, rear +y top, and the rear +x side. See also InGT, InLT and GRMX.
GRMX	Gradient index maximum index. This boundary operand sets the maximum allowable index of refraction value for the gradient index surface defined by **Surf** at the wavelength defined by **Wave**. The index is checked at six places: the front vertex, front +y top, front +x side, rear vertex, rear +y top, and the rear +x side. See also InGT, InLT and GRMN.
GTCE	Glass TCE. This operand returns the Thermal Coefficient of Expansion Alpha1 as listed in the glass catalog for the glass on the surface defined by **Surf**. For non-glass surfaces, see "TCVA".
HACG	Unused.
HHCN	Test for the hyperhemisphere condition. Zemax traces the specified ray at the wavelength defined by **Wave** to the surface defined by **Surf**, and computes the x, y, and z intercept coordinates. Then, the x and y coordinates only are used in the sag expression for that surface to see what z coordinate results. If the z coordinates are not the same, then HHCN returns 1, otherwise it returns zero. This operand can be used to prevent optimizations from reaching solutions that require hyperhemispheric surface shapes. See "Hx, Hy, Px, and Py".

IMAE	Image analysis data. This operand returns fractional efficiency, centroid, or RMS data as computed by the geometric image analysis feature, using whatever the current default settings are, except for "Show" which is always set to spot diagram for this computation, "Surface", which may be selected using **Surf**, "Field", which may be selected using **Field**, and "Field Size", which may be selected using **Field Size**. To use this operand, first define the settings on the geometric image analysis feature as desired, then press Save on the settings box. The operand IMAE will return the desired data as computed by the image analysis feature.

If **Surf** is 0, the surface specified by the saved settings will be used. If **Surf** is greater than zero, then the data will be computed at the specified surface.

If **Field** is 0, the field number specified by the saved settings will be used. If **Field** is greater than zero, then the data will be computed at the specified field.

If **Field Size** is 0, the Field Size specified by the saved settings will be used. If **Field Size** is greater than zero, then the data will be computed using the specified Field Size.

Data is 0 for efficiency, 1 for X-centroid, 2 for Y-centroid, or 3-5 for X-, Y- or Radial-direction RMS, respectively. All values other than efficiency are in lens units.

If **Wave** is 0, the wave number specified by the saved settings file will be used. If **Wave** is greater than zero, then the data will be computed using the specified wavelength number.

See the discussion "Optimizing with the IMAE operand". |
| IMSF | Image Surface. This operand dynamically changes which surface is interpreted as the "image surface" for all subsequent operands. The new image surface is defined by **Surface**. The primary usage for this operand is to optimize image quality at intermediate surfaces. Set **Surface** to 0 to restore the image surface back to the original surface.

If **Refocus?** is 0, the specified surface becomes the new image surface and is not modified in any way. If **Refocus?** is 1, then a dummy plane surface will follow the specified surface, and a paraxial focus solve will be used to place this dummy surface at the focus of the specified surface. This option is most useful for optimizing aberrations at a virtual focus for intermediate surfaces that are not already at focus.

Note this operand only temporarily changes a copy of the lens data use for evaluating the merit function and has no impact on the original lens data. Care should be taken when the selected surface precedes the original stop surface. For information on how this operand is implemented see "Evaluating results at intermediate surfaces". |
| INDX | Index of refraction. Returns the current index at the surface defined by **Surf** at the wavelength defined by **Wave**. |
| InGT | Index "n" greater than. This boundary operand constrains the index of refraction at the wavelength defined by **Wave** of the gradient index surface defined by **Surf** at one of six points inside the gradient index lens. For n=1, the point is the front vertex, n=2 is the |

	front +y top, n=3 is the front +x side, n=4 is the rear vertex, n=5 is the rear +y top, and n=6 is the rear +x side. In all cases the operand bounds the index at the specified point to be greater than the specified target value. For example, "I4GT" constrains the minimum index at the rear vertex of the surface (i.e. the vertex of the next surface) of the gradient index lens. In all cases the +y top and +x side distance is defined by the larger of the front and rear semi-diameters set on the main spreadsheet. See also GRMN and GRMX, which are similar operands that are easier to use.
InLT	Index "n" less than. This operand is similar to InGT except it constrains the maximum value of the index of refraction rather than the minimum. See InGT for a complete description of the parameter "n".
InVA	This operand is similar to InGT except it constrains the current value of the index of refraction. See InGT for a complete description of the parameter "n".
ISFN	Image space F/#. This operand is the paraxial infinite conjugate F/#. See WFNO.
ISNA	Image space numerical aperture. This operand is the paraxial image space na at the defined conjugates. See ISFN.
LACL	Lateral color. For focal systems, this is the y-distance between the paraxial chief ray intercepts of the two extreme wavelengths defined by **Minw** and **Maxw**, measured in lens units. For afocal systems, this is the angle in afocal mode units between the paraxial chief rays of the two extreme wavelengths defined by **Minw** and **Maxw**.
LINV	Lagrange (or optical) invariant of system in lens units at the wavelength defined by **Wave**. The paraxial marginal and chief ray data are used to compute this value.
LOGE	Log base e of an operand. **Op#** is the row number of the operand value to take the log of. If the value is less than or equal to zero, the value zero is returned.
LOGT	Log base 10 of an operand. **Op#** is the row number of the operand value to take the log of. If the value is less than or equal to zero, the value zero is returned.
LONA	Longitudinal aberration, measured in lens units for focal systems and diopters for afocal systems. This is the defocus from the current image to the image at the wavelength defined by **Wave** and pupil zone defined by **Zone**. If **Zone** is zero, paraxial rays are used to determine the paraxial image locations. If **Zone** is greater than 0.0 and less than or equal to 1.0, real marginal rays are used to determine the image locations. In this case, **Zone** corresponds to the Py coordinate of the real marginal ray. See AXCL.

LPTD	This boundary operand constrains the slope of the axial gradient index profile from changing signs within a gradient index component for the surface defined by **Surf**. See the section "Using gradient index operands".
MAXX	Returns the largest value within the indicated range of operands defined by **Op#1** and **Op#2.** See MINN.
MCOG	Multi-configuration operand greater than. This is used to constrain values in the multi-configuration editor. **Op#** defines which multi-configuration operand is used. **Cfg#** defines which configuration is used.
MCOL	Multi-configuration operand less than. This is used to constrain values in the multi-configuration editor. See MCOG.
MCOV	Multi-configuration operand value. This is used to directly target or compute values in the multi- configuration editor. See MCOG.
MINN	Returns the smallest value within the indicated range of operands. See MAXX.
MNAB	Minimum Abbe number. This boundary operand constrains the Abbe number of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MXAB. This operand controls multiple surfaces simultaneously.
MNCA	Minimum center thickness air. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be greater than the specified target value. See also MNCT and MNCG. This operand controls multiple surfaces simultaneously.
MNCG	Minimum center thickness glass. This boundary operand constrains each of the thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be greater than the target value. See also MNCT and MNCA. This operand controls multiple surfaces simultaneously.
MNCT	Minimum center thickness. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MNCG and MNCA. This operand controls multiple surfaces simultaneously.
MNCV	Minimum curvature. This boundary operand constrains each of the curvatures of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MXCV. This operand controls multiple surfaces simultaneously.
MNDT	Minimum diameter to thickness ratio. Controls the minimum allowable value on the ratio

	of surface diameter to center thickness of surfaces from **Surf1** to **Surf2**. Only surfaces with non-unity index of refraction are considered. See also MXDT. This operand controls multiple surfaces simultaneously.
MNEA	Minimum edge thickness air. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be greater than the specified target value. See also MNET, MNEG, ETGT, and XNEA. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XNEA for constraining non-rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MNEG	Minimum edge thickness glass. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be greater than the specified target value. See also MNET, MNEA, ETGT, and XNEG. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XNEG for constraining non-rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MNET	Minimum edge thickness. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** to be greater than the specified target value. See also MNEG, MNEA, ETGT, and XNET. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XNET for constraining non-rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MNIN	Minimum index at d-light. This boundary operand constrains the Nd value of surfaces from **Surf1** and **Surf2** to be greater than the specified target value. See also MXIN. This operands controls multiple surfaces simultaneously.
MNPD	Minimum $\Delta P_{g, F}$. This boundary operand constrains the deviation of the partial dispersion of surfaces between **Surf1** and **Surf2** to be greater than the specified target value. See also MXPD. This operand controls multiple surfaces simultaneously.
MNRE	Minimum real ray angle of exitance. This boundary operand constrains the minimum ray exit angle over a range of surfaces. The angles are measured in degrees between the surface normal and the exiting ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note the angle of exitance is always positive. See also MXRE, MNRI, and MXRI.

	See "Hx, Hy, Px, and Py".
MNRI	Minimum real ray angle of incidence. This boundary operand constrains the minimum ray incidence angle over a range of surfaces. The angles are measured in degrees between the surface normal and the incident ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note the angle of incidence is always positive. See also MNRE, MXRE, and MXRI. See "Hx, Hy, Px, and Py".
MNSD	Minimum semi-diameter. Constrains the semi-diameter to be larger than the specified target over the surface range between **Surf1** and **Surf2**. This operand controls multiple surfaces simultaneously.
MSWA	Modulation square-wave transfer function, average of sagittal and tangential. See MTFA for details.
MSWS	Modulation square-wave transfer function, sagittal. See MTFA for details.
MSWT	Modulation square-wave transfer function, tangential. See MTFA for details.
MTFA	Diffraction modulation transfer function, average of sagittal and tangential. The parameters are: **Samp**: Higher sampling yields higher accuracy at the expense of computation time. To confirm the computation has acceptable accuracy, start at 1, and increment the sampling until the results change by less than the desired accuracy. Note that extreme precision is not required for good optimization results; three significant figures is usually adequate. There are two algorithms available for computing the MTF. If **Grid** is zero (recommended), a fast, sparse sampling integration method is used to compute the MTF. The fast sampling method used by MTFA, MTFS, and MTFT is not directly related to the MTF Analysis feature. Because only a single spatial frequency is required, the method of computation used by the MTF operands is different, and generally much faster, than the algorithm used by the analysis feature. To select the alternate grid-based algorithm used by the MTF analysis feature, set **Grid** equal to 1. The grid-based algorithm is usually slower than the default algorithm if the MTF is reasonably good (greater than 5%), but the grid algorithm converges faster if the aberrations are very large and the resulting MTF is very low. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Freq**: The spatial frequency in MTF units (see "MTF Units"). If the sampling is set too low for accurate computation of the MTF, then the MTF operands all return zero. If both the tangential and sagittal MTF are needed; place the MTFT and MTFS operands on adjacent lines and they will be computed simultaneously. See

	"Performing an optimization".
	Data Type: Specifies the data to be returned. An input of 0 will return the modulation amplitude; an input of 1 will return the real part; an input of 2 will return the imaginary part; an input of 3 will return the phase. This input is only available for the MTFA, MTFS, and MTFT operands (and not the equivalent square-wave operands).
MTFS	Modulation transfer function, sagittal. See MTFA for details.
MTFT	Modulation transfer function, tangential. See MTFA for details.
MTHA	Huygens Modulation transfer function, average of sagittal and tangential. This operand computes the diffraction MTF using the Huygens method (see "Huygens MTF"). The parameters are:
	Samp: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. The sampling is assumed to be the same for both pupil and image.
	Wave: The wavelength number to use (use 0 for polychromatic).
	Field: The field number.
	Freq: The spatial frequency in MTF units (see "MTF Units"). If the sampling is set too low for accurate computation of the MTF, then the MTF operands all return zero.
	Pol?: Set to 0 to ignore polarization and 1 to consider it.
	All Conf?: Set to 0 to use the current configuration (defined by the last CONF operand preceding this operand), and 1 to sum over all configurations. See "Huygens MTF" for a full discussion of this option.
	Ima Delta: The image delta in micrometers used for the computation. If zero, the default image delta is used.
	If both the tangential and sagittal MTF are needed; place the MTHT and MTHS operands on adjacent lines and they will be computed simultaneously. See "Performing an optimization".
MTHS	Huygens Modulation transfer function, sagittal. See MTHA for details.
MTHT	Huygens Modulation transfer function, tangential. See MTHA for details.
MXAB	Maximum Abbe number. This boundary operand constrains the Abbe number of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also MNAB. This operand controls multiple surfaces simultaneously.

MXCA	Maximum center thickness air. This boundary operand constrains each of the thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be less than the target value. See also MXCT and MXCG. This operand controls multiple surfaces simultaneously.
MXCG	Maximum center thickness glass. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be less than the target value. See also MXCT and MXCA. This operand controls multiple surfaces simultaneously.
MXCT	Maximum center thickness. This boundary operand constrains each of the center thicknesses of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also MXCG and MXCA. This operand controls multiple surfaces simultaneously.
MXCV	Maximum curvature. This boundary operand constrains each of the curvatures of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also MNCV. This operand controls multiple surfaces simultaneously.
MXDT	Maximum diameter to thickness ratio. Controls the maximum allowable value on the ratio of surface diameter to center thickness from **Surf1** to **Surf2**. Only surfaces with non-unity index of refraction are considered. See also MNDT. This operand controls multiple surfaces simultaneously.
MXEA	Maximum edge thickness air. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have air (i.e. no glass) as the glass type to be less than the specified target value. See also MXET, MXEG, ETLT, and XXEA. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XXEA for constraining non-rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MXEG	Maximum edge thickness glass. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** which have a non-air glass type to be less than the target value. See also MXET, MXEA, ETLT, and XXEG. This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface only; see XXEG for constraining non- rotationally symmetric surfaces. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
MXET	Maximum edge thickness. This boundary operand constrains each of the edge thicknesses of surfaces from **Surf1** to **Surf2** to be less than the specified target value. See also "MXEG", "MXEA", "ETLT", and "XXET". This operand controls multiple surfaces simultaneously. The boundary applies to the top "+y" edge of the surface

| | only; see XXET for constraining non-rotationally symmetric surfaces.

Zone, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter. |
|---|---|
| MXIN | Maximum index at d-light. This boundary operand constrains the Nd value of surfaces between **Surf1** and **Surf2** to be less than the specified target value. See also MNIN. This operand controls multiple surfaces simultaneously. |
| MXPD | Maximum $\Delta P_{g,F}$. This boundary operand constrains the deviation of the partial dispersion of surfaces between **Surf1** and **Surf2** to be less than the specified target value. See also MNPD. This operand controls multiple surfaces simultaneously. |
| MXRE | Maximum real ray angle of exitance. This boundary operand constrains the maximum ray exit angle over a range of surfaces. The angles are measured in degrees between the surface normal and the exiting ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note the angle of exitance is always positive. See also MNRE, MNRI, and MXRI. See "Hx, Hy, Px, and Py". |
| MXRI | Maximum real ray angle of incidence. This boundary operand constrains the maximum ray incidence angle over a range of surfaces. The angles are measured in degrees between the surface normal and the incident ray at the surface range defined by **Surf1** through **Surf2** at the wavelength defined by **Wave**. If **Wave** is zero the primary wavelength is used. Note the angle of incidence is always positive. See also MNRE, MNRI, and MXRE. See "Hx, Hy, Px, and Py". |
| MXSD | Maximum semi-diameter. Constrains the semi-diameter to be less than the specified target over the surface range from **Surf1** to **Surf2**. |
| NORD | Normal distance to the next surface. This operand computes the surface normal vector at the coordinate defined by **X** and **Y** on the surface defined by **Surf**, then returns the distance to the next surface measured along the normal vector. |
| NORX | Normal vector x component. This operand returns the x component of the surface normal vector at the coordinate defined by **X** and **Y** on the surface defined by **Surf**. If **Global** is zero, the vector is in surface local coordinates, if **Global** is 1, the vector is in global coordinates. |
| NORY | Normal vector y component. This operand returns the y component of the surface normal vector at the coordinate defined by **X** and **Y** on the surface defined by **Surf**. If **Global** is zero, the vector is in surface local coordinates, if **Global** is 1, the vector is in global coordinates. |
| NORZ | Normal vector z component. This operand returns the z component of the surface |

	normal vector at any coordinate defined by **X** and **Y** on the surface defined by **Surf**. If **Global** is zero, the vector is in surface local coordinates, if **Global** is 1, the vector is in global coordinates.
NPGT	Non-sequential parameter greater than. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Object** defines the object number in the NSC group. The parameter number is defined by **Param**.
NPLT	Non-sequential parameter less than. See NPGT.
NPVA	Non-sequential parameter value. See NPGT.
NPXG	Non-sequential object position x greater than. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Object** defines the object number in the NSC group. If **Ref?** is 0, the coordinates are relative to the reference object. If **Ref?** is 1, the coordinates are relative to the origin of the NSC coordinate system or entry port. If **Ref?** is 2, then the coordinates are relative to the global coordinate reference surface.
NPXL	Non-sequential object position x less than. See NPXG.
NPXV	Non-sequential object position x value. See NPXG.
NPYG	Non-sequential object position y greater than. See NPXG.
NPYL	Non-sequential object position y less than. See NPXG.
NPYV	Non-sequential object position y value. See NPXG.
NPZG	Non-sequential object position z greater than. See NPXG.
NPZL	Non-sequential object position z less than. See NPXG.
NPZV	Non-sequential object position z value. See NPXG.
NSDC	Non-sequential coherent data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector. If **Pix#** is a positive integer, then the data from the specified pixel is returned. If **Pix#** is

	zero, then the sum of the data for all pixels for that detector object is returned.
	Data is 0 for real, 1 for imaginary, 2 for amplitude, and 3 for power. See "User defined operands" for complete details.
NSDD	Non-sequential incoherent intensity data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector. If **Det#** is zero, then all detectors are cleared. If **Det#** is less than zero, then the detector defined by the absolute value of **Det#** only is cleared. Note that one NSDD operand clearing *all* detectors must be present prior to any subsequently defined NSDD operands. For Detector Rectangles, Detector Surfaces, and all faceted detectors: If **Pix#** is a positive integer greater than zero, then the data from the specified pixel is returned. Otherwise, the following data is returned depending upon the value of **Pix#**: 0: Sum of the total flux in position space, average flux/area in position space, or total flux in angle space for all pixels for that detector object, for **Data** = 0, 1, or 2, respectively. -1: Maximum flux or flux/area. -2: Minimum flux or flux/area. -3: Number of rays striking the detector. -4: Standard deviation (RMS from the mean) of all the *non-zero* pixel data. -5: The mean value of all the *non-zero* pixel data. -6, -7, -8: The x, y, or z coordinate of the position or angle Irradiance or Intensity centroid, respectively. -9,-10, -11, -12, -13: The RMS radius, x, y, z, or xy cross term distance or angle of all the pixel data with respect to the centroid. These are the second moments r^2, x^2, y^2, z^2, and xy, respectively. **Data** is 0 for flux, 1 for flux/area, 2 for flux/solid angle pixel, and 3 for normalized flux. Only values 0 and 1 (for flux and flux/area) are supported for faceted detectors. A value of 3 is only supported when **Pix#** is a positive integer; in this case the returned flux for the pixel is normalized to the peak flux for all consecutive NSDD operands in the Merit Function Editor which refer to the same surface and detector. A value of 3 should generally be used only as a part of the NSC Bitmap Merit Function (see "NSC Bitmap Merit Function Tool"). For Detector Rectangles only: If **Pix#** is not a positive integer, then a number of edge pixels may be ignored in the calculation if desired. The number of edge pixels that will be ignored is given by the **# Ignored** input. For non-zero values of **# Ignored**, the number of pixels accounted for in the calculation will be reduced along all boundaries of the detector. For example, if a detector has 100 pixels in X and 200 pixels in Y and the **# Ignored** value is 2, then the calculation will be based upon detector data from 96 pixels in X and 196 pixels in Y, with data from the last 2 edge pixels along the left, right, bottom and top boundaries all ignored. The only calculation not affected by the **# Ignored** input is the number of rays striking the detector, as given by **Pix #** = -3 (in

	addition to all values calculated for positive integer inputs to **Pix #**).

For Detector Volumes: **Pix#** is interpreted as the voxel number. For **Data** values of 0, 1, or 2, the returned value is incident flux, absorbed flux, or absorbed flux per unit volume, respectively. If **Pix#** is zero, the value returned is the sum for all pixels.

For Detector Color objects, use NSDE instead. For Detector Polar objects, use NSDP. |
| NSDE | Non-sequential Detector Color object data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector. If **Det#** is zero, then all detectors are cleared. If **Det#** is less than zero, then the detector defined by the absolute value of **Det#** only is cleared.

If **Pix#** is a positive integer greater than zero, then the data from the specified pixel is returned. Otherwise, the following data is returned depending upon the value of **Pix#**: 0: Sum of all data for all pixels on the detector. For chromaticity coordinates this is the average over all pixels in the detector that have non-zero lumen values. If **Pix#** is zero the <area> values described below are the area of the entire detector, measured in analysis units for position space data or solid angle in steradians for angle space data.

-1: Maximum data.

-2: Minimum data.

-3: Number of rays striking the detector.

Angle? is 0 for position space data, 1 for angle space data.

Data is an integer code that defines the type of data to return, defined as follows:

1: power in <sprefix> watts where <sprefix> is the source units prefix.

2: power/area in <aprefix> watts per <area> where <aprefix> is the analysis units prefix. If **Angle?**

is 0 <area> is the area in analysis units. If **Angle?** is 1 then <area> is solid angle in steradians.

3: power in <sprefix> lumens where <sprefix> is the source units prefix.

4: power/area in <aprefix> lumens per <area> where <aprefix> is the analysis units prefix. If **Angle?**

is 0 <area> is the area in analysis units. If **Angle?** is 1 then <area> is solid angle in steradians.

5/6: CIE 1931 chromaticity coordinate x/y.

7/8: CIE 1976 chromaticity coordinate u'/v'.

9/10/11: CIE 1931 tristimulus X/Y/Z value in lumens per <area>. If **Angle?** is 0 <area> is the area in analysis units. If **Angle?** is 1 then <area> is solid angle in steradians.

12/13: Color Rendering Index (CRI) and Correlated Color Temperature (CCT), the |

	latter value in Kelvin. These values are only returned if the option to "Record Spectral Data" has been selected for the Detector Color object (only available in Zemax-Premium) and if the **Pix#** and **Angle?** are both zero. The returned values are based on the variation of flux with wavelength over the entire detector.
	14/15/16: Normalized CIE 1931 tristimulus X/Y/Z value. Identical to the values returned by Data 9, 10, and 11, but normalized to the peak tristimulus Y value for all consecutive NSDE operands in the Merit Function Editor which refer to the same surface and detector. Generally used only as a part of the NSC Bitmap Merit Function (see "NSC Bitmap Merit Function Tool"). Only valid if **Pix#** is a positive integer and **Angle?** is zero.
	If **Pix#** is not a positive integer, then a number of edge pixels may be ignored in the calculation if desired. The number of edge pixels that will be ignored is given by the **# Ignored** input. For non- zero values of **# Ignored**, the number of pixels accounted for in the calculation will be reduced along all boundaries of the detector. For example, if a detector has 100 pixels in X and 200 pixels in Y and the **# Ignored** value is 2, then the calculation will be based upon detector data from 96 pixels in X and 196 pixels in Y, with data from the last 2 edge pixels along the left, right, bottom and top boundaries all ignored. Calculations that are not affected by the **# Ignored** input are the number of rays striking the detector, as given by **Pix #** = -3, and the CRI and CCT values, as given by Data = 12 and Data = 13, respectively (the **# Ignored** input also does not affect any values calculated for positive integer inputs to **Pix #**).
	If the source units are joules, then the data 1 and 2 are in joules and data 3 and 4 are in talbots. See "Units". See also NSDD and NSDP.
NSDP	Non-sequential Detector Polar object data. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Det#** refers to the object number of the desired detector. If **Det#** is zero, then all detectors are cleared. If **Det#** is less than zero, then the detector defined by the absolute value of **Det#** only is cleared.
	If **Pix#** is a positive integer greater than zero, then the data from the specified pixel is returned. Otherwise, the following data is returned depending upon the value of **Pix#**:
	0: Sum of all data for all pixels on the detector. For chromaticity coordinates this is the average over all pixels in the detector that have non-zero lumen values.
	-1: Maximum data.
	-2: Minimum data.
	-3: Number of rays striking the detector.
	-4: The radial RMS in degrees of the distribution with respect to 0 degrees. If **Pix#** is set equal to -
	4, then the only valid inputs for **Data** are 1 or 3. All other inputs will return a value of zero.
	Data is an integer code that defines the type of data to return, defined as follows:
	1: power in <sprefix> watts where <sprefix> is the source units prefix.

	2: power/solid angle in <sprefix> watts per steradian where <sprefix> is the source units prefix.
	3: power in <sprefix> lumen where <sprefix> is the source units prefix.
	4: power/solid angle in <sprefix> lumens per steradian where <sprefix> is the source units prefix.
	5/6: CIE 1931 chromaticity coordinate x/y.
	7/8: CIE 1976 chromaticity coordinate u'/v'.
	9/10/11: CIE 1931 tristimulus X/Y/Z value in lumens/steradian.
	If the source units are joules, then the data 1 and 2 are in joules and data 3 and 4 are in talbots. See "Units". See also NSDD and NSDE.
NSLT	Non-sequential LightningTrace. This operand traces mesh rays as defined by the LightningTrace analysis. **Surf** is the surface number of the Non-Sequential surface. **Src#** refers to the object number of the desired source from which to trace mesh rays. If **Src#** is zero, rays will be traced from all sources. **Ray Samp** defines the sampling to use in defining the ray mesh: 0 = Low (1X), 1
	= 4X, 2 = 16X, 3 = 64X, 4 = 256X, 5 = 1024X. **Edge Samp** defines the sampling to use for resolving the ray mesh around the edges of objects, with the same interpretation for the numeric inputs as for **Ray Samp** (but with inputs only going from 0 to 4). **RT?** defines whether mesh rays should be traced (when **RT?** = 0) or whether all analysis rays should be traced using conventional ray tracing (when **RT?** = 1). If **RT?** = 1, then this operand will act just like the NSTR operand with polarization, splitting, and scattering all turned off.
	An NSDD, NSDE, or NSDP operand with Det# set equal to 0 must be included in the merit function prior to this operand.
	See "The LightningTrace Control" for more information. See also NSTR.
NSRA	Non-sequential single ray trace. **Src#** refers to the object number of the desired source. This source must be defined to trace just a single analysis ray. If **Splt?** is non-zero, then splitting is on. If **Pol?** is non-zero then polarization will be used. If splitting is on polarization is automatically selected.
	Scattering is always turned off for this feature, since scattering introduces random paths for the child rays, which is not suitable for optimization. Errors are always considered.
	If multiple NSRA operands trace the same ray are adjacent in the merit function editor, the ray will only be traced once for efficiency.
	Seg# refers to the segment number that contains the data to be returned. Use -1 for the last segment.
	Data refers to the data type for the specified segment.
	Use **Data** values 1-9 for x-coordinate, y-coordinate, z-coordinate, x-cosine, y-cosine, z-

	cosine, x- normal, y-normal, and z-normal, respectively. These values are relative to the entry port, see 31- 39 below.

Use **Data** values 10-15 for path-to, intensity, phase of, phase at, index, and starting phase, respectively. Note that the phase at value is not modulo 2*pi, as is the case with ZRD files.

Use **Data** values 16-17 for the sum of the path, or optical path, respectively, in lens units from the source to the end of the specified segment. These values do not include the phase of diffractive surfaces.

Use **Data** values 21-26 for Ex real, Ex imaginary, Ey real, Ey imaginary, Ez real, and Ez imaginary, respectively. Polarization must be on for real and imaginary data to be returned.

Use **Data** value 27-29 for the phase of Ex, Ey, and the phase difference between Ex and Ey, respectively. All values are in radians. Polarization must be on for valid data to be returned.

Use **Data** values 31-39 for the coordinate data defined by **Data** values 1-9 converted to coordinates relative to the global coordinate reference surface.

Source # refers to the element number to use if the source is an array; see "Modeling arrays of sources" for the numbering scheme.

For more information on these data items see "Ray database (ZRD) files". |
| NSRM | Non-sequential Rotation Matrix component. **Surf** defines the surface number of the NSC group (always 1 in pure NSC systems). **Object** defines the object number in the NSC group.

If **Ref?** is 0, the coordinates are relative to the reference object. The rotation matrix will always be the identity matrix when using this reference.

If **Ref?** is 1, the coordinates are relative to the origin of the NSC coordinate system or entry port. If **Ref?** is 2, then the coordinates are relative to the global coordinate reference surface.

The 3 x 3 R matrix has 9 components. If **Data** is 1, NSRM returns R[1][1], if **Data** is 2, NSRM returns

R[1][2], etc... through **Data** = 9 returning R[3][3]. |
| NSRW | Non-sequential roadway lighting data.

Data defines the desired roadway lighting data to be returned. Use 0 for average luminance, 1 for overall uniformity of luminance, 2 for longitudinal uniformity of luminance, 3 for threshold increment,

4 for surround ratio, 5 for average illuminance, 6 for minimum illuminance, and 7 for uniformity of

horizontal illuminance. The arguments **# Lanes, Lane Width, Spacing,** and **Offset** |

	are all as
	defined for the roadway lighting analysis. **Arrange** defines the luminaire arrangement along the roadway. Use 0 for single-side, 1 for double-sided, and 2 for staggered. **Surf. Class** defines the
	road surface classification. Use 0 for R1, 1 for R2, 2 for R3, and 3 for R4.
	See "Roadway Lighting" for more information.
	The first NSRW with unique settings must be immediately preceded by an NSTW operand in order to compute data. NSRW operands with the same settings (except **Data**) must be placed on adjacent lines. These subsequent operands will simply return data cached by the first NSRW operand. This operand is intended to be used as part of the NSC Roadway Merit Function Tool.
	See "NSC Roadway Merit Function Tool" for more information.
NSST	Non-sequential single ray trace. This operand traces a single sequential ray through the system to any specified Non-Sequential surface, and returns various data about the ray within the Non- Sequential surface. **Surf** is the surface number of the Non-Sequential surface. **Wave** is the wavelength number. For a definition of **Hx**, **Hy**, **Px**, and **Py**, see "Hx, Hy, Px, and Py". **Data** determines what data NSST will compute and return as follows:
	0, 1, 2: The x, y, or z coordinate at the ray-object intercept point.
	3, 4, 5: The x, y, or z direction cosines of the ray after reflection/refraction from the surface.
	6, 7, 8: The x, y, or z direction cosines of the ray up to the ray-object intercept point.
	9, 10, 11: The surface normal at the ray-object intercept point.
	12: The face number of the object the ray struck.
	Object specifies which object the desired data is for.
	In the general case, a ray may strike the same object multiple times. By default, NSST will return the data for the *last* intercept on the specified object. To select a specific intercept, add 1000 to the **Data** value for each time the ray strikes the object. For example, to compute the y coordinate value of the ray on the third time the ray strikes an object, use a **Data** value of 3001. If the ray does not strike the object, or does not strike the object the specified number of times, the operand returns a value of zero, but no warning or error is issued.
	All coordinates and cosines are in the coordinate system of the Non-Sequential surface. An NSDC, NSDD, NSDE, or NSDP operand with Det# set equal to 0 must be included in the merit function prior to this operand. See also NSTR.
NSTR	Non-sequential trace. **Src#** refers to the object number of the desired source. If **Src#** is zero, all sources will be traced. If **Splt?** is non-zero, then splitting is on. If **Scat?** is non-zero, then scattering is on. If **Pol?** is non-zero then polarization will be used. If

	splitting is on polarization is automatically selected. If **IgEr?** is non-zero, then errors will be ignored. See "Optimizing with sources and detectors in non-sequential mode" for complete details.
	An NSDD, NSDE, or NSDP operand with Det# set equal to 0 must be included in the merit function prior to this operand.
	See also NSST.
NSTW	Non-sequential roadway lighting raytrace.
	This operand is solely for optimizing roadway lighting applications in conjunction with the NSRW operand and the NSC Roadway Merit Function. If **Splt?** is non-zero, then splitting is on. If **Scat?** is non-zero, then scattering is on. If **Pol?** is non-zero then polarization will be used. If splitting is on polarization is automatically selected. If **IgEr?** is non-zero, then errors will be ignored. **Origin** is the row number of the origin object. **MH** is the mounting height used in the calculation.
	See "Roadway Lighting" for more information.
	NSTW should be immediately followed by an NSRW operand to compute data. This operand is intended to be used as part of the NSC Roadway Merit Function Tool.
	See "NSC Roadway Merit Function Tool" for more information.
NTXG	Non-sequential object tilt about x greater than. See NPXG.
NTXL	Non-sequential object tilt about x less than. See NPXG.
NTXV	Non-sequential object tilt about x value. See NPXG.
NTYG	Non-sequential object tilt about y greater than. See NPXG.
NTYL	Non-sequential object tilt about y less than. See NPXG.
NTYV	Non-sequential object tilt about y value. See NPXG.
NTZG	Non-sequential object tilt about z greater than. See NPXG.
NTZL	Non-sequential object tilt about z less than. See NPXG.
NTZV	Non-sequential object tilt about z value. See NPXG.
OBSN	Object space numerical aperture. This is only useful for finite conjugate systems, and is calculated on axis at the primary wavelength.

OOFF	This operand indicates an unused entry in the operand list. OOFF operands are automatically converted to BLNK operands upon evaluation of the merit function. OOFF is only used to indicate that the merit function operand type was not recognized.
OPDC	Optical path difference with respect to chief ray in waves at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py" .
OPDM	Optical path difference with respect to the mean OPD over the pupil at the wavelength defined by **Wave**. OPDM has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py".
OPDX	Optical path difference with respect to the mean OPD over the pupil with tilt removed at the wavelength defined by **Wave**. OPDX has the same restrictions that TRAC does; see TRAC for a detailed discussion. See "Hx, Hy, Px, and Py".
OPGT	Operand greater than. This is used to make the value of the operand defined by **Op#** greater than the target value.
OPLT	Operand less than. This is used to make the value of the operand defined by **Op#** less than the target value.
OPTH	Optical path length. This is the distance, in lens units, the specified ray travels to the surface defined by **Surf** at the wavelength defined by **Wave**. The distance is measured from the object for finite conjugates; otherwise the distance is referenced to the first surface. The optical path accounts for the index of refraction of the media, and for phase adding surfaces such as gratings and binary optics. See PLEN. See "Hx, Hy, Px, and Py".
OPVA	Operand value. This operand constrains the value of the operand defined by **Op#** to be equal to the target value.
OSCD	Offense against the sine condition (OSC) at the wavelength defined by **Wave**. There are two definitions for OSC supported. The first definition is as described in Welford, Aberrations of Optical Systems (see "REFERENCES ON LENS DESIGN"). This definition is used if **Zone** is zero. An alternate definition due to Prof. Roland Shack which supports computation of OSC as a function of pupil zone and uses only real rays is available. This definition is used if **Zone** is not zero. In this case, **Zone** corresponds to the Py coordinate of the real marginal ray. The two methods will give very similar results for systems with modest F/#'s and aberrations when **Zone** is 1.0 for the alternate definition. This operand has no meaning if the system is not axially symmetric.

OSUM	Sums the values of all operands between the two operands defined by **Op#1** and **Op#2**. See SUMM.
PANA	Paraxial ray x-direction surface normal at the ray-surface intercept at the wavelength defined by **Wave**. This is the x component of the surface normal vector at the intersection point of the specified paraxial ray and the surface defined by **Surf**, in the local coordinate system. See "Hx, Hy, Px, and Py".
PANB	Paraxial ray y-direction surface normal at the ray-surface intercept at the wavelength defined by **Wave**. This is the y component of the surface normal vector at the intersection point of the specified paraxial ray and the surface defined by **Surf**, in the local coordinate system. See "Hx, Hy, Px, and Py".
PANC	Paraxial ray z-direction surface normal at the ray-surface intercept at the wavelength defined by **Wave**. This is the z component of the surface normal vector at the intersection point of the specified paraxial ray and the surface defined by **Surf**, in the local coordinate system. See "Hx, Hy, Px, and Py".
PARA	Paraxial ray x-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PARB	Paraxial ray y-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PARC	Paraxial ray z-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py .
PARR	Paraxial ray radial coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. This is the radial distance from the local axis to the intersection of the surface defined by **Surf** and the specified paraxial ray, in the local coordinate system. See "Hx, Hy, Px, and Py".
PARX	Paraxial ray x-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PARY	Paraxial ray y-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by

	Wave. See "Hx, Hy, Px, and Py".
PARZ	Paraxial ray z-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PATX	Paraxial ray x-direction ray tangent. This is the tangent of the angle the paraxial ray makes in the X-Z plane after refraction from surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PATY	Paraxial ray y-direction ray tangent. This is the tangent of the angle the paraxial ray makes in the Y-Z plane after refraction from surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
PETC	Petzval curvature in inverse lens units at the wavelength defined by **Wave**. Not valid for non-paraxial systems.
PETZ	Petzval radius of curvature in lens units at the wavelength defined by **Wave**. Not valid for non- paraxial systems.
PIMH	Paraxial image height at the paraxial image surface at the wavelength defined by **Wave**. Not valid for non-paraxial systems.
PLEN	Path length. This operand computes the total optical path length (including index of refraction and phase surfaces) between surfaces **Surf1** and **Surf2** for the specified ray, which is always traced at the primary wavelength. PLEN is essentially the difference between two OPTH operands. See OPTH. See "Hx, Hy, Px, and Py".
PMAG	Paraxial magnification. This is the ratio of the paraxial chief ray height on the paraxial image surface to the object height at the wavelength defined by **Wave**. Only useful for finite conjugate systems. Note the paraxial image surface is used even if the system is not at paraxial focus.
PMGT	Parameter greater than. This boundary operand constrains the value of the parameter defined by **Param** for the surface defined by **Surf** to be greater than the target value. The parameter values have different meanings depending upon the surface type. See the Chapter "Surface Types" for a description of the parameter values.
PMLT	Parameter less than. This boundary operand constrains the value of the parameter defined by **Param** for the surface defined by **Surf** to be less than the target value. The parameter values have different meanings depending upon the surface type. See the Chapter "Surface Types" for a description of the parameter values.

PMVA	Parameter value. This boundary operand constrains the value of the parameter defined by **Param** for the surface defined by **Surf** to be equal to the target value. The parameter values have different meanings depending upon the surface type. See the Chapter "Surface Types" for a description of the parameter values.
PnGT	This operand is obsolete, use PMGT instead.
PnLT	This operand is obsolete, use PMLT instead.
PnVA	This operand is obsolete, use PMVA instead.
POPD	Physical Optics Propagation Data. For important details see "PHYSICAL OPTICS PROPAGATION".
	To use this operand, first define the settings on the POP analysis feature as desired, then press
	Save on the settings box. The operand will return data based upon the selected settings.
	If **Surf** is zero, then the saved ending surface number will be used; otherwise, the specified surface will be used as the ending surface. If **Wave** is zero, then the saved wavelength number will be used; otherwise, the specified wavelength number will be used. If **Field** is zero, then the saved field number will be used; otherwise, the specified field number will be used.
	Data determines what data the POP feature will compute and return as follows:
	0: The total fiber coupling. This is the product of the system efficiency and the receiver efficiency.
	1: The system efficiency for fiber coupling.
	2: The receiver efficiency for fiber coupling.
	3: The total power.
	4: The peak irradiance.
	5, 6, 7: The pilot beam position, Rayleigh range, beam waist (x).
	8, 9, 10: The pilot beam position, Rayleigh range, beam waist (y).
	11, 12, 13: The local X, Y, Z coordinates of the center of the beam array on the end surface (this is a reference point and is not related to the amplitude of the beam).
	21, 22: The X, Y coordinates of the centroid of the intensity distribution in local coordinates relative to the center of the beam.
	23, 24, 25, 26: The X, Y beam width and X, Y M-squared values, respectively. See

	"Beam width and M-squared". 27, 28, 29: The second moment of the beam for x^2, y^2, and xy respectively. 30, 31, 32: The mean, RMS and PTV irradiance variation, respectively, of the non-zero amplitude portion of the beam. These operands should only be used when the beam has nearly uniform irradiance and has just been clipped by a surface aperture. 33, 34, 35: The mean, RMS and PTV phase variation in radians, respectively, of the non-zero amplitude portion of the beam. These operands should only be used when the beam has nearly uniform irradiance and has just been clipped by a surface aperture. 40, 41, 42: The fraction of the total power enclosed within a circle of radius specified by the **Xtr1** value in lens units, referenced to the beam centroid (40), chief ray (41), or surface vertex (42). 50, 51, 52: The radius in lens units of the circle at which the fraction of the total power enclosed is equal to the value specified by the **Xtr1** value. The circle is centered on the beam centroid (50), chief ray (51), or surface vertex (52). 60, 61, 62, 63: The fiber coupling receiver efficiency amplitude and phase in radians for the Ex field (60 and 61) and the Ey field (62 and 63). These values do not consider system efficiency (see **Data** type 1 above). Undefined **Data** values will return 0. The **Xtr1** and **Xtr2** values are only used by selected data numbers which are reserved for future expansion of this feature. If adjacent POPD operands all have the same **Surf**, **Wave**, **Field**, **Xtr1**, and **Xtr2** values, then the POP analysis is done just once and all data returned at one time. Note the POPD operands must be on adjacent rows in the MFE for this efficiency to be implemented.
POPI	Physical Optics Propagation Data. For important details see "PHYSICAL OPTICS PROPAGATION". To use this operand, first define the settings on the POP analysis feature as desired, then press Save on the settings box. The operand will return data based upon the selected settings. If **Surf** is zero, then the saved ending surface number will be used; otherwise, the specified surface will be used as the ending surface. If **Wave** is zero, then the saved wavelength number will be used; otherwise, the specified wavelength number will be used. If **Field** is zero, then the saved field number will be used; otherwise, the specified field number will be used. **Data** determines what data the POP feature will compute and return as follows: 0, 1, 2: The total (Ex + Ey), Ex only, or Ey only Irradiance.

	3, 4, 5, 6: Ex-real, Ex-Imaginary, Ey-real, Ey-imaginary.
	7, 8: Ex, Ey phase in radians. Undefined **Data** values will return 0.
	The **Pix#** refers to the desired pixel of the beam. The pixel number is greater than or equal to zero, and less than nx*ny, where nx and ny are the number of columns and rows, respectively. The pixel number is generally defined as p = x + y*nx where x is the integer row number and y is the integer column number, and 0 <= x < nx and 0 <= y < ny.
	If adjacent POPI operands all have the same **Surf**, **Wave**, and **Field** values, then the POP analysis is done just once and all data returned at one time. Note the POPI operands must be on adjacent rows in the MFE for this efficiency to be implemented.
POWF	Power at a field point. Computes the power or effective focal length (EFL) after refraction from the surface defined by **Surf** at the wavelength defined by **Wave** for any point in the field. For the **Data** parameter, use 0 for spherical, 1 for cylinder, 2 for max, 3 for min, 4 for tangential, 5 for sagittal, 6 for y, 7 for x, and 8 for astigmatic power in diopters. To get the EFL values, add 9 to these codes. For example, for tangential EFL use **Data** = 13. For cylinder, the EFL value is the difference between the maximum and minimum focal lengths. For astigmatic, the EFL is the difference between the x and y direction focal lengths (x-y).

For a full description of this type of analysis, see "Power Field Map". See "Hx, Hy, Px, and Py". |
| POWP | Power at a point in the pupil. Computes the power or effective focal length (EFL) after refraction from the surface defined by **Surf** at the wavelength defined by **Wave** for any point in the field, at any point in the pupil. For the **Data** parameter, use 0 for spherical, 1 for cylinder, 2 for max, 3 for min, 4 for tangential, 5 for sagittal, 6 for y, 7 for x, and 8 for astigmatic power in diopters. To get the EFL values, add 9 to these codes. For example, for tangential EFL use **Data** = 13. For cylinder, the EFL value is the difference between the maximum and minimum focal lengths. For astigmatic, the EFL is the difference between the x and y direction focal lengths (x-y).

For a full description of this type of analysis, see "Power Pupil Map". See "Hx, Hy, Px, and Py". |
POWR	The surface power (in inverse lens units) of the surface defined by **Surf** at the wavelength defined by **Wave**. This operand only works for standard surfaces.
PRIM	Primary wavelength. This is used to change the primary wavelength number to the wavelength defined by **Wave** during merit function evaluation. This operand does not use the target or weight columns.
PROB	Multiplies the value of the operand defined by **Op#** by the factor defined by **Factor**.
PROD	Product of two operands (**Op#1** X **Op#2**). See PROB.

QOAC	Unused.
QSUM	Quadratic sum. This operand squares and then adds all operands between the two operands defined by **Op#1** and **Op#2**, then takes the square root of the sum. See also SUMM, OSUM, EQUA.
RAED	Real ray angle of exitance. This is the angle in degrees between the surface normal and the ray after refraction or reflection for the surface defined by **Surf** at the wavelength defined by **Wave**. See also RAID. See "Hx, Hy, Px, and Py".
RAEN	Real ray angle of exitance. This is the cosine of the angle between the surface normal and the ray after refraction or reflection at the surface defined by **Surf** at the wavelength defined by **Wave**. See also RAIN. See "Hx, Hy, Px, and Py".
RAGA	Global ray x-direction cosine. This is the direction cosine of the ray in the global coordinate system at the surface defined by **Surf** at the wavelength defined by **Wave**. The origin of the global coordinate system is at the global reference surface. See "Hx, Hy, Px, and Py".
RAGB	Global ray y-direction cosine. See RAGA.
RAGC	Global ray z-direction cosine. See RAGA.
RAGX	Global ray x-coordinate. This is the coordinate in lens units in the global coordinate system at the surface defined by **Surf** at the wavelength defined by **Wave**. The origin of the global coordinate system is at the global reference surface. See "Hx, Hy, Px, and Py".
RAGY	Global ray y-coordinate. See RAGX.
RAGZ	Global ray z-coordinate. See RAGX.
RAID	Real ray angle of incidence. This is the angle in degrees between the surface normal and the incident ray at the surface defined by **Surf** at the wavelength defined by **Wave**. Note the angle of incidence is always positive. See also RAED. See "Hx, Hy, Px, and Py".

RAIN	Real ray angle of incidence. This is the cosine of the angle between the surface normal and the ray before refraction at the surface defined by **Surf** at the wavelength defined by **Wave**. See also RAEN. See "Hx, Hy, Px, and Py".
RANG	Ray angle in radians with respect to z axis. The angle is measured with respect to the local Z axis of the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAA	Real ray x-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py" .
REAB	Real ray y-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAC	Real ray z-direction cosine of the ray after refraction from the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAR	Real ray radial coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAX	Real ray x-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAY	Real ray y-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REAZ	Real ray z-coordinate in lens units at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RECI	Returns the reciprocal of the value of operand **Op#1**. See also "DIVI".

RELI	Relative illumination. This operand computes the relative illumination of the field point defined by **Field** relative to the (0, 0) field point. Note that in some systems, the illumination increases off axis, and for these systems the RI may be greater than one since the RELI operand uses (0, 0) as a reference field point. Vignetting factors are always removed for this calculation. The other parameters are: **Samp**: The grid size. A value of 10 would yield a 10 x 10 grid of rays. **Wave**: The wavelength number to use. **Pol?**: Set to 0 to ignore polarization and 1 to consider it. See also EFNO.
RENA	Real ray x-direction surface normal at the ray-surface intercept at the surfaced defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RENB	Real ray y-direction surface normal at the ray-surface intercept at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RENC	Real ray z-direction surface normal at the ray-surface intercept at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RETX	Real ray x-direction ray tangent (slope) at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
RETY	Real ray y-direction ray tangent (slope) at the surface defined by **Surf** at the wavelength defined by **Wave**. See "Hx, Hy, Px, and Py".
REVR	Reverse Radiance. This operand returns the Total Flux computed by the reverse radiance feature (see "ReverseRadiance™ Analysis"). The **Surf** value is 1 for non-sequential systems, otherwise, it is the surface number of the non-sequential group. **Object** is the number of the Reverse Detector Object, **Pixel #** is 0 to sum all or the individual pixel number, and Data Type is 0 through 3 for total flux or tristimulus X, Y, Z. This operand works independently and does not require the use of other operands for ray tracing or clearing detectors.
RGLA	Reasonable glass. This operand restricts the deviation the index, Abbe, and deviation of the partial dispersion values may take from actual glasses in the currently loaded glass catalogs. See "Optimizing zoom and multi-configuration lenses" for a complete

	discussion including a description of **Wn**, **Wa** and **Wp**. The constraint is active over the surface range specified by **Surf1** and **Surf2**.
RSCE	RMS spot radius with respect to the centroid in lens units. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RSCH	RMS spot radius with respect to the chief ray in lens units. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RSRE	RMS spot radius with respect to the centroid in lens units. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RSRH	RMS spot radius with respect to the chief ray in lens units. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py".
RWCE	RMS wavefront error with respect to the centroid in waves. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py", and "OPTIMIZATION REFERENCE POINTS".
RWCH	RMS wavefront error with respect to the chief ray in waves. This operand uses a Gaussian quadrature method that is accurate for systems with unvignetted circular pupils. **Ring** is used to specify the number of rings of rays traced. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py" , and "OPTIMIZATION REFERENCE POINTS".
RWRE	RMS wavefront error with respect to the centroid in waves. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified

	wavelength number will be used. See "Hx, Hy, Px, and Py", and "OPTIMIZATION REFERENCE POINTS".
RWRH	RMS wavefront error with respect to the chief ray in waves. This operand uses a rectangular grid of rays to estimate the RMS. This operand considers vignetting. A **Samp** value of n will trace an n x n grid per pupil quadrant. If **Wave** is zero; then a wavelength weighted polychromatic calculation is performed; otherwise, the specified wavelength number will be used. See "Hx, Hy, Px, and Py", and "OPTIMIZATION REFERENCE POINTS".
SAGX	The sag in lens units of the surface defined by **Surf** at X = the semi-diameter, and Y = 0. See also SSAG.
SAGY	The sag in lens units of the surface defined by **Surf** at Y = the semi-diameter, and X = 0. See also SSAG.
SCUR	Surface Curvature. Computes curvature and related data of the surface defined by **Surf** at the coordinate defined by **X** and **Y**. In all cases below, the "maximum" data is determined by calculation of the curvature at 50 points equally spaced from the vertex of the surface to the specified **X** and **Y** coordinate, and the largest absolute value found is returned. If **data** is 0-3, the returned value is the tangential, sagittal, tangential-sagittal, or maximum tangential-sagittal curvature, respectively. If **data** is 4-7, the returned value is the x, y, x-y, or maximum x-y curvature, respectively. If **data** is 8-9, the returned value is the absolute value of (R*Sc), where R is the radial coordinate and Sc is the sagittal curvature, or the maximum of this value, respectively.
SDRV	Surface Derivative. Computes the first or second derivative of the surface sag (along the local Z axis) of the surface defined by **Surf** at the coordinate defined by **X** and **Y**. If **data** is 0 or 1, the returned value is the first derivative in the tangential or sagittal direction. If **data** is 2 or 3, the returned value is the second derivative in the tangential or sagittal direction.
SFNO	Sagittal working F/#, computed at the field point defined by **Field** and the wavelength defined by **Wave**. See TFNO.
SINE	Sine of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.

SKIN	Skip if not symmetric. See SKIS.
SKIS	Skip if symmetric. If the lens is rotationally symmetric, then computation of the merit function continues at the operand defined by **Op#**.
SMIA	SMIA-TV Distortion. **Field** is the zero distortion reference field position, or use zero to indicate the field position (0, 0). **Wave** is the wavelength number, or use zero for the primary wavelength. **X- Width** and **Y-Width** are the full field width in field units. For more information see "SMIA-TV Distortion".
SPCH	Spherochromatism in lens units. This is the difference between the real marginal axial color and the paraxial axial color of the two extreme wavelengths defined by **Minw** and **Maxw**. The distance is measured along the Z axis. **Zone** defines the zone for which the real marginal axial color is computed. **Zone** corresponds to the Py coordinate of the real marginal ray. Not valid for non-paraxial systems.
SPHA	Spherical aberration in waves contributed by the surface defined by **Surf** at the wavelength defined by **Wave**. If **Surf** is zero, the sum for the entire system is used. This is the third order spherical aberration calculated from the Seidel coefficients, and is not valid for non-paraxial systems.
SQRT	Square root of the operand defined by **Op#**.
SSAG	The sag in lens units of the surface defined by **Surf** at the coordinate defined by **X** and **Y**. See also SAGX, SAGY.
STHI	Surface Thickness. This operand computes the thickness of the surface defined by **Surf** to the next surface at the coordinate defined by **X** and **Y** on the surface. The calculation accounts for the sag and center thickness of the surface and the sag of the next surface, but not any tilts and decenters between the surfaces.
STRH	Strehl Ratio. This operand computes the Strehl Ratio using the Huygens PSF computation (see "Huygens PSF"). The parameters are: **Samp**: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. The sampling is assumed to be the same for both pupil and image. **Wave**: The wavelength number to use (use 0 for polychromatic). **Field**: The field number. **Pol?:** Set to 0 to ignore polarization and 1 to consider it. **All Conf?:** Set to 0 to use the current configuration (defined by the last CONF operand preceding this operand), and 1 to sum over all configurations. See "Huygens PSF"

	for a full discussion of this option. By default, the image delta used in the Huygens PSF computation for the evaluation of this operand is 2x smaller than the nominal default image delta. See "Huygens PSF" for the formula used to calculate the nominal default image delta. This is done to zoom in on the peak of the PSF, allowing for a more accurate computation of the Strehl ratio. As a result, the value reported will, in general, be different than the value reported using the nominal default image delta in the Huygens PSF analysis.
SUMM	Sum of two operands (**Op#1 + Op#2**). See OSUM.
SVIG	Sets the vignetting factors for the current configuration. The **Precision** is either 0, 1, or 2; for high, medium, or low precision, respectively. The SVIG operand can take a significant time to execute at high precision, although the resulting optimization is generally superior. Note that the operands SVIG and CVIG only modify the vignetting factors for evaluating subsequent optimization operands. When the end of the merit function is reached, the vignetting factors will be restored to their original values. See also "CVIG".
TANG	Tangent of the value of the operand defined by **Op#**. If **Flag** is 0, then the units are radians, otherwise, degrees.
TCGT	Thermal Coefficient of expansion greater than. This boundary operand constrains the TCE of the surface defined by **Surf** to be greater than the specified target value.
TCLT	Thermal Coefficient of expansion less than. This boundary operand constrains the TCE of the surface defined by **Surf** to be less than the specified target value.
TCVA	Thermal Coefficient of expansion value. This boundary operand constrains the TCE of the surface defined by **Surf** to be equal to the specified target value. For glass surfaces, see "GTCE".
TFNO	Tangential working F/#, computed at the field point defined by **Field** and the wavelength defined by **Wave**. See SFNO.
TGTH	Sum of glass thicknesses from **Surf1** to **Surf2**. Note that the sum is inclusive, it is not the thickness between the two surfaces. See TTHI.
TMAS	Total mass. Computes the mass of the glass lenses within the range of surfaces from **Surf1** to **Surf2**. The mass of a surface considers the volume enclosed up to the following surface; therefore to compute the mass of a single element the first and last surface numbers should be the same. See "Comments on computing element volumes" for a discussion of how element masses and volumes are computed.

TOLR	Tolerance data. **Data** is 0 for RSS estimated change in performance, 1 for nominal performance, and 2 for estimated performance (nominal plus estimated change). **File** is the integer number corresponding to the tolerance settings file to use. **Config#** is -2 for the configuration defined by the last CONF operand, -1 for all configurations, 0 to use the configuration set in the tolerance settings file and 1 or greater for a specific configuration number. For details, see "Optimizing tolerance sensitivity" .
TOTR	Total track (length) of lens in lens units. See "Total track".
TRAC	Transverse aberration radial direction measured in image space with respect to the centroid for the wavelength defined by **Wave**. Unlike most other operands, TRAC critically depends upon the placement of other TRAC operands within the Merit Function Editor to work correctly. TRAC operands must be grouped together by field position and wavelength. Zemax traces all TRAC rays with a common field point together, and then uses the collective data to compute the centroid of all the rays. Each ray individually is then referenced to the computed centroid. This operand should only be entered into the Merit Function Editor by the Sequential Merit Function tool, and is not recommended for use directly by the user. See "Hx, Hy, Px, and Py".
TRAD	The x component of the TRAR only. TRAD has the same restrictions that TRAC does; see TRAC for a detailed discussion.
TRAE	The y component of the TRAR only. TRAE has the same restrictions that TRAC does; see TRAC for a detailed discussion.
TRAI	Transverse aberration radius measured at the surface defined by **Surf** at the wavelength defined by **Wave** with respect to the chief ray. Similar to TRAR, except a surface other than the image surface may be specified. See "Hx, Hy, Px, and Py".
TRAN	Unused.
TRAR	Transverse aberration radial direction measured in image space at the wavelength defined by **Wave** with respect to the chief ray. See ANAR. See "Hx, Hy, Px, and Py".
TRAX	Transverse aberration x direction measured in image space at the wavelength defined by **Wave** with respect to the chief ray. See "Hx, Hy, Px, and Py".

TRAY	Transverse aberration y direction measured in image space at the wavelength defined by **Wave** with respect to the chief ray. See "Hx, Hy, Px, and Py".
TRCX	Transverse aberration x direction measured in image space with respect to the centroid. TRCX has the same restrictions that TRAC does; see TRAC for a detailed discussion.
TRCY	Transverse aberration y direction measured in image space with respect to the centroid. TRCY has the same restrictions that TRAC does; see TRAC for a detailed discussion.
TTGT	Total thickness greater than. This boundary operand constrains the total thickness, including surface sags of the surface defined by **Surf** and the immediately following surface at their respective semi-diameter values, to be greater than the specified target value. The thickness is calculated along the +y axis if **Code** is zero, the +x axis if **Code** is 1, the -y axis if **Code** is 2, and the -x axis if **Code** is 3. This operand automatically changes the sign on thicknesses in mirror spaces to always yield a positive value for physically possible lenses. See TTLT and TTVA.
TTHI	Sum of thicknesses of surfaces from **Surf1** to **Surf2**. Note that the sum is inclusive, it is not the thickness between the two surfaces. See TGTH.
TTLT	Total thickness less than. See TTGT.
TTVA	Total thickness value. See TTGT.
UDOP	User defined operand. Used for optimizing numerical results computed in externally compiled programs. See "User defined operands". See also ZPLM.
USYM	If present in the merit function, this operand instructs Zemax to assume radial symmetry exists in the lens even if Zemax detects symmetry does not exist. This speeds execution of the merit function in some special cases. See "Assume Axial Symmetry".
VOLU	Volume of element(s) in cubic cm. Computes the volume of the lenses and air spaces for the range of surfaces defined by **Surf1** and **Surf2**. The volume of a surface includes the volume enclosed up to the following surface; therefore to compute the volume of a single element the first and last surface numbers should be the same. See "Comments on computing element volumes" for a discussion of how element masses and volumes are computed.
WFNO	Working F/#. See "Working F/#", and ISFN, SFNO, and TFNO.
WLEN	Wavelength. This operand returns the wavelength defined by **Wave** in micrometers.

XDGT	Extra data value greater than. The **Edv#** must be between 1 and 200 to indicate which of the extra data values is selected for the surface defined by **Surf**.
XDLT	Extra data value less than. The **Edv#** must be between 1 and 200 to indicate which of the extra data values is selected for the surface defined by **Surf**.
XDVA	Extra data value. The **Edv#** must be between 1 and 200 to indicate which of the extra data values is selected for the surface defined by **Surf**.
XENC	Extended source encircled energy (distance). This operand computes the distance in micrometers to the specified fraction of extended source geometric encircled energy, using whatever the current default settings are. To use this operand, first define the settings on the extended source encircled energy feature as desired, then press Save on the settings box. The only settings that are overwritten are those set by **Type** and **Wave**. **Type** is 1 for encircled, 2 for x only, 3 for y only, 4 for ensquared, 5 for x distribution, and 6 for y distribution. **Wave** is the input wavenumber. **Frac** is the fraction of energy desired, and must be between zero and 1, exclusive. **Frac** is ignored for type 5 or 6; for these types the returned valued is the full width half max independent of **Frac**. **Max Radius** is the maximum distance in radial distance in micrometers. If this value is zero, the default setting is used. See also XENF, DENC, DENF, GENC, and GENF.
XENF	Extended source encircled energy (fraction). This operand computes the fraction of extended source geometric encircled, ensquared, x only, or y only (enslitted) energy at a given distance from the reference point. The **Type** is 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. **Wave** is the input wavenumber. The options and settings are identical to XENC, except **Dist**, which here is used as the distance at which the fraction of energy is desired. See also XENC, GENC, GENF, DENC, and DENF.
XNEA	Minimum edge thickness for the range of air surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are at least the minimum specified thickness. This operand controls multiple surfaces simultaneously. See MNEA. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of

	0.5 would compute the thickness at 0.5 times the semi-diameter.
XNEG	Minimum edge thickness for the range of glass surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are at least the minimum specified thickness. This operand controls multiple surfaces simultaneously. See MNEG. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XNET	Minimum edge thickness for the range of surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are at least the minimum specified thickness. This operand controls multiple surfaces simultaneously. See MNET. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XXEA	Maximum edge thickness for the range of air surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are no more than the maximum specified thickness. This operand controls multiple surfaces simultaneously. See MXEA. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XXEG	Maximum edge thickness for the range of glass surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are no more than the maximum specified thickness. This operand controls multiple surfaces simultaneously. See MXEG. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.
XXET	Maximum edge thickness for the range of surfaces defined by **Surf1** and **Surf2**. This operand checks the edge thickness at numerous points around the perimeter of the surface, and tests if all points are no more than the maximum specified thickness. This operand controls multiple surfaces simultaneously. See MXET. **Zone**, if non zero, scales the radial aperture at which the thickness is computed. A **Zone** value of 0.5 would compute the thickness at 0.5 times the semi-diameter.

YNIP	YNI-paraxial. This number is the product of the paraxial marginal ray height times the index times the angle of incidence at the surface defined by **Surf** at the wavelength defined by **Wave**. This quantity is related to the narcissus contribution of the specified surface. See Applied Optics, Vol. **21**, 18, p3393.
ZERN	Zernike Fringe coefficient. The parameters are:
	Term: The Zernike term number (1 - 37 for fringe, 1 - 231 for standard or annular).
	Wave: The wavelength number.
	Samp: The pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc.
	Field: The field number.
	Type: The Zernike type (0 for fringe, 1 for standard, 2 for annular). **Epsilon**: The obscuration ratio (for annular coefficients only). **Vertex?**: If 1, the OPD is referenced to the surface vertex.
	The **Term** value, if negative or zero, may also be used to return other data from the Zernike fitting as follows:
	-8: Peak to Valley OPD (to centroid)
	-7: Peak to Valley OPD (to chief)
	-6: RMS to zero reference (unused by Zemax)
	-5: RMS to chief ray
	-4: RMS to centroid
	-3: Variance
	-2: Strehl Ratio
	-1: RMS fit error
	0: Maximum single point fit error
	Note that if you use multiple ZERN operands which only differ in the **Term** value, they should be placed on adjacent lines in the editor so Zemax only does the fitting once; otherwise, the computation is slower.
	Multiple Zernike terms are always used in the fitting process, even if only one coefficient is requested. The maximum number of terms used in the computation depends on the **Type** and the **Term** settings. The minimum number of terms used for all types is 11. This means that **Term** is only used if set greater than or equal to 11. If **Type** is standard or annular, the maximum term number computed is set equal to the largest term requested by any **Term** value in adjacent ZERN operands.
ZPLM	Used for optimizing numerical results computed in ZPL macros. See "User defined

	operands". See also UDOP.
ZTHI	This operand controls the variation in the total thickness of the range surfaces defined by **Surf1** and **Surf2** over multiple configurations. It is similar to the TTHI operand, except it is an inequality operator. The target value specified is the maximum allowed difference between the TTHI at each defined configuration. For example, if there are 3 configurations where TTHI 3 8 would evaluate to

17, 19, and 18.5, respectively, ZTHI will return 2 (i.e. 19-17) if the target is less than 2. Otherwise, ZTHI returns the target value. To keep all zoom configurations the same length, use a target of 0. |

Notes on Operand Weights

The operational operands (SUMM, OSUM, DIFF, PROD, DIVI, SQRT) along with the parametric operands (CVGT, CVLT, CTGT, CTLT, etc.) can be used to define very general and complex optimization operands, as discussed in the section "Defining complex operands".

Because of the dimensional differences between parameters such as effective focal length (tens of millimeters or more) and RMS spot radius (micrometers), usually a weighting of one is sufficient for quantities measured in lens units. However, the residual value of the effective focal length with this weighting is not likely to be zero. Increasing the weighting will bring the resulting system closer to the desired effective focal length. This effect is often noticeable when defining ETGT (edge thickness greater than) operands. Usually ETGT with a target of zero will often yield a value just slightly less than zero. Rather than increase the weight, it is much simpler to provide a target value of 0.1, or some such number.

After making changes to the operand list, the current values of each operand can be updated using the Update button in the toolbar. This is also useful for checking to see what the current values of each operand are, and which has the greatest contribution to the merit function. The percent contribution is defined as

$$\% \text{ contrib}_i = 100 \times \frac{W_i (V_i - T_i)^2}{\sum_j W_j (V_j - T_j)^2} .$$

where the index j indicates the sum over all operands.

The merit function is automatically saved with the lens file.

Understanding boundary operands

The boundary operands such as MNCT, CTGT, DIMX, and others behave somewhat differently than specific target operands such as TRAR and REAY. When you specify a boundary on a parameter, you specify the target value as the definition of the boundary. For example, to maintain a minimum center thickness on surface 5 of 10 mm, you might use a command such as CTGT 5 10 (where the 5 is in the Int1 column and 10 is in the target column). If you update the merit function and then observe the "value" column of that operand, there are two possibilities for the value: 1) if the boundary is violated, that is, the center thickness is less than 10, then the actual value of the thickness will be displayed, or 2) if the boundary is not violated, that is, the center thickness is greater than 10, then the value 10 will be displayed.

The rule is if the boundary is violated, the actual value is shown; if it is not violated, the value is set to the target and is therefore ignored by the optimization algorithm. If during optimization the boundary becomes violated, then the value will automatically be updated and the optimization algorithm will attempt to correct the offending parameter.

The boundary operands which constrain a range of surfaces are slightly more complicated. These multiple surface operands return values which represent the total effect of all violated boundaries within the specified surface range. For example, the operand MNCT 1 10 will constrain the minimum center thickness of surfaces 1 through 10. If the target is 3.0, which defines the boundary, then the difference between the value of the operand and the target is the sum of the difference between 3.0 and the thicknesses of all surfaces between 1 and 10 whose center thickness is less than 3.0. If only one surface in the range has a center thickness less than 3.0, say 2.5, then the operand has a value of 2.5. If a second surface is added that has a thickness less than 3.0, say 2.2, then the operand will have a value of 1.7 (2.5 minus .8; the .8 is 3.0 - 2.2). The total difference between the target of the operand and the value is 3.0 - 1.7 or 1.3. This difference of 1.3 is due to the violation of 0.5 by the first surface and another 0.8 by the second surface.

If calculating the value of these boundary operands seems confusing, don't worry; Zemax does all the calculations for you. All you need to do is to specify the boundary type (such as MNCT or MNET) the boundary range (surface 1 through 10, or whatever) and the desired value (3 mm or whatever). If all the boundary constraints are met, then the operand value is equal to the target, otherwise, the value will be different and the merit function will increase. The increased merit function will cause the optimization algorithm to seek a reduction of the operand contribution.

If a boundary operand does not seem to work, there are several things to check:

1) Make sure the variables you have defined can have some effect on the boundary operands. A common mistake is to specify MNCT and have some "frozen" thickness within the surface range. If the thickness violates the boundary and it is not variable, Zemax can't fix it. The operands DO NOT ignore violated but frozen boundaries.

2) If there is a small residual error, try increasing the boundary value. For example, if MNCT is used with a target of 0.0, and the value is a small number (like -.001) the problem is not that the operand doesn't work, it is that the residual error is too small to increase the merit function significantly. It is usually better to increase the target to 0.1, or some other number, rather than to increase the weight. Increasing the weight will only lead to a smaller violation (like -.0000001) rather than meeting the boundary.

3) Check to see if there is a reasonable contribution to the merit function. You can easily check this with the percent contribution column. By looking at the percent contribution column, verify that the operand in question has enough influence on the total merit function. If it does not, increase the weight, or see the preceding paragraph for advice on changing the target.

Understanding the boundary operands is a crucial part of mastering Zemax optimization, and with a little practice you will find them to offer excellent control and flexibility.

Performing an optimization

To begin optimization, click the Optimize! Button in the Automatic Optimization section of the Optimize Tab.

For more infromation, the "Optimize!" section of The Optimize Tab.

Defining complex operands

Although the default merit function, coupled with a few predefined operands, is perfectly suitable for the majority of optical designs, there are times when an unusual constraint needs to be added to the merit function. Rather than define a very large number of very specific operands, Zemax allows you to build your own operands out of simple building blocks.

Zemax allows very general operand definitions. There are two tricks to creating these operands. First, use certain operands with zero-weighting to define the parameters you need, and second, use the operational operands to define relationships between them. For example, suppose you require that the thickness of surface 3 and the thickness of surface 4 sum to 10. There is an operand that does this, TTHI. The command structure would look like this:

Number	Type	Int1	Int2	Target	Weight
1	TTHI	3	4	10	1

However, for illustration only, note that there is an alternate way of calculating the same thing:

Number	Type	Int1	Int2	Target	Weight
1	CTVA	3		0	0
2	CTVA	4		0	0
3	SUMM	1	2	10	1

Operand 1 uses the Center Thickness VAlue (CTVA) command to extract the value of the thickness of surface 3. Similarly, operand 2 is used to extract the thickness of surface 4. The zero weighting on both operands means the optimization algorithm ignores the constraint; it is only used as an intermediate step. Operand 3 now sums two operands: number 1 and number 2. The result is the sum of the thicknesses of surfaces 3 and 4 is the value of operand 3, and this has a non-zero weight. The optimization algorithm will attempt to drive the sum to 10.

Why go to all the trouble of this three-step process if a single TTHI command would do the same thing? The reason is that this approach can be extended to develop very general operands. For example, suppose you wanted the radius of curvature of surface 5 to be centered on the vertex of surface 8. Study the following commands to see if you understand how this is done:

Number	Type	Int1	Int2	Target	Weight
1	CVVA	5		0	0
2	TTHI	5	7	0	0
3	PROD	1	2	1	1

The CVVA command extracts the curvature of surface 5, the curvature we want to control. TTHI 5 7 calculates the distance from surface 5 to surface 8 (note we only sum to surface 7 to get to surface 8, since the thickness of surface 8 gives the distance to surface 9). Since the curvature of the surface is the reciprocal of the radius, the product of the curvature and the distance must be one; hence the target is 1 for operand 3. Operand 3 is also the only weighted operand in the sequence.

Now consider the requirement that the thickness of surface 5 must be greater than twice the radius of curvature of surface 4 plus the conic constant of surface 2 (this is nonsensical, but illustrative of the flexibility in the approach):

Number	Type	Int1	Int2	Target	Weight
1	CTVA	5		0	0
2	CVVA	4		0	0
3	CONS			2	0
4	DIVI	3	2	0	0
5	COVA	2		0	0
6	SUMM	4	5	0	0
7	DIFF	1	6	0	0
8	OPGT	7		0	1

Operand 1 extracts the (center) thickness of surface 5. Operand 2 extracts the value of the curvature of surface 4. Operand 3 sets a constant of two, and operand 4 divides the value 2 by the curvature (yielding twice the radius of curvature). COVA extracts the conic, and SUMM ads operands 2 and 4. Operand 7 takes the difference of the thickness and twice the radius plus the conic. Since we want the former to exceed the latter, we set an operand greater than constraint; the only one to have a non-zero weighting.

Optimizing zoom and multi-configuration lenses

Optimizing zoom lenses is virtually identical to optimizing conventional single-configuration lenses. See "Multi-Configurations" for details.

Optimizing tolerance sensitivity

The idea of optimizing a lens for reduced tolerance sensitivity is that a very low merit function is not useful if the as-built performance is significantly degraded by tolerance sensitivity. Therefore, the optimal design has a reasonable nominal performance but is relatively insensitive to manufacturing defects. The theory is there may be a potential trade-off between performance and tolerance sensitivity.

Tolerance sensitivity in an optical design comes from many sources, including angles of incidence of rays on surfaces, aberration balancing, and the nature of the potential fabrication defects. The interaction of multiple defects makes accurate tolerance prediction a difficult statistical problem. For a complete discussion of tolerance analysis, see the chapter "TOLERANCING".

The optimization operand TOLR can in principle be used to optimize for reduced sensitivity to tolerances. To use TOLR, first optimize a design for reasonable starting performance. Then, define the relevant tolerance operands, limits, compensators, and criterion as described in the chapter on Tolerancing. Save the options of the tolerance dialog box; Zemax uses these saved options to compute the data returned by TOLR. TOP files are created by saving the tolerance options from the tolerance dialog. To use the most recently saved tolerance settings, set the file number on the TOLR operand to zero. To use a specific saved settings file, use an integer value between 1 and 999. The saved settings file name must be of the format TOLRnnn.TOP, where nnn is the 3 digit integer specified on the TOLR operand file argument. For example, if the desired tolerance settings are saved in a file named TOLR005.TOP, the integer file value

should be 5. Note only files with the correctly formatted name may be referenced and used by TOLR. The TOP files must be stored in the <data>\Configs folder (see "Folders"). The only criterion that is not allowed is "Merit Function" since that would create an infinite loop. If a script tolerance criterion is being used, do not load any merit functions within the script that contain TOLR operands, as this may also create an infinite loop that Zemax cannot detect.

As part of the tolerance sensitivity analysis, Zemax computes a nominal performance estimate, and predicts an RSS estimated change. The predicted total performance is the sum of nominal and estimated change. These values are computed and returned by the TOLR operand for optimization. TOLR values may be targeted and weighted as any other optimization operand.

TOLR takes a single "data" integer. If data is 0, TOLR returns the estimated change in performance. If data is 1, TOLR returns the nominal performance. If data is 2, TOLR returns the predicted performance, which is the sum of the nominal and estimated performance change. The first instance of TOLR (the first row in which TOLR appears) is the only instance for which the tolerance analysis is actually run, so that TOLR only runs once for each merit function evaluation. The other values computed by TOLR are saved and can be retrieved by using data values other than 0 on subsequent calls to TOLR. Other values of data are currently unused and may be used for future expansion of this feature.

The practical difficulty of optimization of tolerance sensitivity lies in the computation time. Complicated lenses may have hundreds of tolerance operands, and complex criterion may take considerable time to compute. When using TOLR, the sensitivity analysis may be computed thousands or even millions of times, which can make the use of TOLR impractical. The following tips should be used to improve the efficiency of the tolerance analysis and make TOLR efficient:

-Test and verify the tolerance sensitivity analysis outside of optimization before using TOLR.

-Use "Sensitivity" and not "Inverse Sensitivity". Inverse sensitivity is slower and meaningless for optimization.

-Use "Paraxial Focus" for the "Comp" control if possible... optimization of the compensator values requires an "optimization within an optimization" that is slow and memory intensive; although it is supported if required.

-Zemax will automatically ignore the Monte Carlo portion of the tolerance analysis.

-Do not use "Merit Function" as the tolerance criterion, as this will create an infinite loop.

The faster the tolerance sensitivity runs, the faster TOLR and thus the optimization will execute.

User defined operands

There are times when very complex calculations need to be performed, and the results of the computation need to be optimized. Zemax supports some of these calculations already, such as the MTFA operand which traces many rays, computes the MTF, and then returns a single resulting number to the Merit Function Editor "value" column. Some limited calculations can be performed within the merit function itself; see for example the discussion in the section "Defining complex operands".

However, there are problems for which only the flexibility of a user defined program is sufficient for defining the data computed by an operand. There are two ways of achieving this:

1) Through the use of a ZPL macro

2) Through the use of an externally defined and compiled program

The use of ZPL macros is simpler, well integrated with Zemax, and requires very little programming experience. ZPL macro optimization is generally a better choice for simpler macros that execute fairly quickly.

Externally defined programs are more complex to program, require an external C or other language compiler, and at least some programming experience. However, externally defined programs can be vastly more complex than what is supported by the ZPL macro language, and usually run faster. Complex calculations will benefit more from being externally compiled than simple calculations.

Both the ZPL and the externally compiled methods of implementing UDO's are described in detail below.

Optimizing with ZPL macros

If the ZPL macro language (see "ZEMAX PROGRAMMING LANGUAGE") is sufficient to perform the required computations, then the operand ZPLM may be used to call a ZPL macro from within the merit function. The macro performs the required computations, then returns the result using the ZPL OPTRETURN keyword.

ZPLM is simple to use. The Mac# and Data values are used to specify the macro number and data field number, respectively. The macro number is used to indicate which ZPL macro should be executed, while the data field number indicates which value computed by the macro should be optimized.

The macro number must be an integer between 0 and 99. If the Mac# value is 17, for example, then the macro to be executed must be named ZPL17.ZPL. The macro name must always use a two digit representation of the macro number. If the macro number was 6, then the macro to be executed would be ZPL06.ZPL. The ZPL macro file must reside in the folder for ZPL macros; see "Folders".

The data field number may be any number between 0 and 50, inclusive. This number refers to a position in a global array associated with the lens in memory. During execution of the macro, the macro keyword OPTRETURN specifies which data field number stores the results of the macro calculation. There are 51 different data fields, so that a single macro call can be used to optimize up to 51 different values simultaneously. For example, suppose you needed a macro which computed the total length of the lens from surface 1 to the image surface (this is in effect a user-defined version of the TOTR operand). The macro might look like this:

```
n = NSUR()
x = 0
FOR i = 1, n, 1
x = x + THIC(i)
NEXT
OPTRETURN 0, x
```

Note the use of the OPTRETURN keyword. This keyword stores the resulting value for "x" in the global array position 0. Suppose this macro was named ZPL15.ZPL. To optimize the resulting value for x, the ZPLM merit function operand would be added to the Merit Function Editor, with Mac# = 15 and Data = 0. After updating the merit function, the "value" would be the same as that returned by TOTR, and it can be optimized in the same way.

ZPLM also permits the use of the data in the Merit Function Editor columns. These data fields can be read by the ZPL macro using the PVHX, PVHY, PVPX, and PVPY ZPL functions, respectively. "PV" is a mnemonic for "Pass Value".

There is one very important thing to know about the data field number. If it is zero, then the macro is executed and the value from OPTRETURN 0 is returned. However, if the data field number is not zero, then the macro is not executed, but any previous value stored from an earlier call to the macro is used instead. The advantage to this convention is substantial. If the macro computes many values, all of which need to be optimized, the macro only needs to be called once, yet multiple ZPLM operands can access the data. This is much more efficient than calling the macro multiple times.

For example, suppose a macro named ZPL11.ZPL computes three values, all of which require optimization. In the macro, the values are stored using OPTRETURN:

```
OPTRETURN 0, x
OPTRETURN 1, y
OPTRETURN 2, z
```

Then three ZPLM operands in the merit function can extract the data and perform the optimization with a single call to the macro:

```
ZPLM 11 0
ZPLM 11 1
ZPLM 11 2
```

The macro ZPL11.ZPL is only called during the evaluation of the ZPLM 11 0 operand. Note the data columns can only be used if the Int2 value is zero, since only in this case is the macro evaluated.

Changes made to the lens from within the ZPLM macro

The merit function is always evaluated using a temporary copy of the lens. After evaluation of the merit function, the copy of the lens, and any changes made to the lens, are discarded. For this reason, no changes should be made to the lens data from within the macro called by the ZPLM operand. These changes are not retained and may interfere with the computation of operands following the ZPLM operand in the same merit function evaluation. Zemax does not restore the lens being evaluated to the state it was in prior to the evaluation of the ZPLM specified macro. If however the macro is intentionally used to alter the lens data prior to evaluation of subsequent operands, two macros should be executed. The first should modify the data as required, and the second should restore the data to the original condition. Both macros can be listed in the merit function editor, with the intervening operands executing on the altered lens data.
Macro commands that manipulate the user interface, such as CLOSEWINDOW, WINL(), and GETT() are not effective. The reason for this restriction is that these commands pull information from the single system copy reflected in the user interface. Thus, even though the ZPLM macro updates the system copies, the primary system copy that the user interface displays will not be updated.

ZPLM and the default merit function

ZPLM should not be used in the middle of a default merit function, but should instead be placed either prior to or after the portion of the merit function that Zemax defined by default.

Optimizing with externally compiled programs

The second method of creating a user defined operand (UDO) is to write an external Windows program which computes the data, then use Dynamic Data Exchange (DDE) to pass the data to and from Zemax. The DDE interface in Zemax is documented and described in "ZEMAX EXTENSIONS". The material presented there is not duplicated here; this discussion assumes the material in that chapter is understood.

The operand UDOP is used to call an external client program from within the merit function. The client program performs the required computations, possibly by making multiple DDE calls back to the Zemax server, then returns the result to Zemax using the DDE interface. The computed data is then placed in the "value" column of the Merit Function Editor and thus may be optimized in the usual way.

UDOP is simple to use. The **Prog#** and **Data** values are used to specify the client program number and data field number, respectively. The client program number is used to indicate which client program should be executed, while the data field number indicates which value computed by the client program should be optimized.

The client program number must be an integer between 0 and 99. If **Prog#** is 17, for example, then the client program number to be executed must be named UDO17.EXE. The client program name must always use a two digit representation of the client program number. If the client program number was 6, then the client program to be executed would be UDO06.EXE. The client program file must reside in the <data>\UDO folder (see "Folders").

When reaching a UDOP operand with a data field number of zero (more on the data field number shortly), Zemax will call the client program. The client program is called with the following syntax, assuming the client program number was 17:

UDO17.EXE buffercode Data1 Data2 Data3 Data4

where Data1 through Data4 are the first four data columns from the editor. The buffercode is an integer value provided by Zemax to the client that uniquely identifies the correct lens. Because Zemax is capable of evaluating multiple lenses simultaneously, the buffercode is used as an identifier so that when the client requests or returns data, it is associated with the correct lens. Note that while optimizing Zemax may be evaluating dozens of lenses simultaneously, each slightly different as derivatives are evaluated and the optimization progresses. The client must compute data for the indicated lens.

Once execution in the client has begun, the client program must do the following critical steps:

1) Establish a DDE link with the Zemax server.

2) Load the correct lens in the Zemax server's memory.

3) Compute the required data.

4) Pass the data back to Zemax.

5) Clear the Zemax server's memory.

6) Terminate the DDE link and exit.

The DDE link is typically maintained by the ZCLIENT code, which is described in the DDE chapter (the user is of course free to write their own, if preferred). ZCLIENT calls the UserFunction defined by the user to compute the operand data.

To load the correct lens into the Zemax server's memory, a single item must be sent to the Zemax server, GetUDOSystem. The syntax is "GetUDOSystem, buffercode". This will cause Zemax to retrieve the correct lens from system memory, and all subsequent DDE calls will be for actions (such as ray tracing) on this lens.

The data is then computed, optionally using any of the DDE item calls defined in the DDE chapter. Once the data is computed, up to 1001 values may be sent back to the server, and ultimately to the optimizer within Zemax, with the SetUDOItem command. The syntax is "SetUDOItem, buffercode, data_field, data". All data items must be free formatted integer, exponential, or floating point numbers.

It is critical that the client program pass back a string using SetUDOItem. Zemax will wait for the client program to pass this string back. Zemax has no way of knowing how long the computation may take; and so Zemax will "hang" until it receives the data. If the client program crashes, or never returns the data, Zemax will never complete execution of the operand, and will hang forever. Pressing the escape key from within Zemax will "break" the evaluation and cause Zemax to skip the operand evaluation.

Terminating the link is achieved by the client program UserFunction returning control to ZCLIENT. A sample code that computes 3 data items, called a, b, and c, might look like this:

```
void UserFunction(char *szCommandLine)
{
double a, b, c;
char szBuffer[5000], szSub[256];
int buffer_code;

/* get the buffer code that identifies the lens */
buffer_code = atoi(GetString(szCommandLine, 0, szSub));

/* set the correct lens in the server's memory */
sprintf(szBuffer,"GetUDOSystem,%i", buffer_code);
PostRequestMessage(szBuffer, szBuffer,",",",",");

/* Here is where we compute the data... these lines are omitted */

/* now return the data a, b, and c in data_field 0, 1, and 2 */
sprintf(szBuffer,"SetUDOItem,%i, 0, %.9E", buffer_code, a);
PostRequestMessage(szBuffer, szBuffer);
sprintf(szBuffer,"SetUDOItem,%i, 1, %.9E", buffer_code, b);
PostRequestMessage(szBuffer, szBuffer);
sprintf(szBuffer,"SetUDOItem,%i, 2, %.9E", buffer_code, c);
PostRequestMessage(szBuffer, szBuffer);

/* close the communication, this tells Zemax we are done sending data. */
sprintf(szBuffer,"CloseUDOData,%i", buffer_code);
PostRequestMessage(szBuffer, szBuffer);

}
```

Note that data values may be returned to the client program using SetUDOItem. This is where the data_field number comes into play. The data field number may be any number between 0 and 1000, inclusive. This number refers to a position in a global array associated with the lens in memory. During execution of the client program, the client DDE item "SetUDOItem" is used to return numerical data values. These values are stored in the lens buffer for later use by UDOP.

There are 1001 different data fields, so that a single client program call can be used to optimize up to 1001 different values simultaneously. The data field number indicates which of the returned values should be placed in the "value" column for that UDOP operand.

UDOP also permits the use of the Data1 through Data4 fields. These data fields can be read by the client program, because they are passed on the command line after the buffercode.

There is one very important thing to know about the data_field number when used on the Merit Function Editor UDOP operand. If it is zero, then the client program is executed and the value from data position 0 is placed in the value column. However, if the data field number is not zero, then the client program is not executed, but any previous value stored from an earlier call to the client program is used instead. The advantage to this convention is substantial. If the client program computes many values, all of which need to be optimized, the client program only needs to be called once, yet multiple UDOP operands can access the data. This is much more efficient than calling the client program multiple times.

For example, suppose a client program named UDO25.EXE computes three values, all of which require optimization. In the client program, the values are passed back using SetUDOItem. Then three UDOP operands in the merit function can extract the data and perform the optimization with a single call to the client program:

```
UDOP 25 0
```

```
UDOP 25 1
UDOP 25 2
```

The client program UDO25.EXE is only called during the evaluation of the UDOP 25 0 operand. Note the Data1 through Data4 values can only be used if the data field value is zero, since only in this case is the client program evaluated.

The Merit Function Editor also supports a timeout argument for UDOP. This value sets the maximum amount of time in milliseconds Zemax should wait for a UDOP to compute the requested data and return control to Zemax. The default value of zero will set the timeout to 50,000 milliseconds. Values larger than 50,000 may be used if needed, but it is of course difficult for optimization to make substantial progress if the evaluation of the merit function takes such a long time. Values shorter than 50,000 may also be used to improve the response of Zemax if the UDOP is not working properly. Note that it is normally the case that UDOP data is computed quickly and the timeout limit is only used to handle unexpected error conditions or unusually long computations. If the UDOP completes the requested computation in less time than the timeout limit, the control still returns to Zemax immediately - there is no need or advantage to set the timeout to a small number for UDOP functions that return data quickly.

Unlike using ZPL macros, UDO's may freely change the lens data during evaluation since all DDE commands are executed on a copy of the lens, not the actual lens being optimized.

There is a sample UDO source code file called UDO_DEMO.C; it may be compiled and linked in with ZCLIENT. The executable needs to be renamed UDOxx.EXE where xx is a two digit integer. The sample UDO returns 6 values: Data1, Data2, Data3, Data4, and two dummy constants, in data positions 0 through 5, respectively.

Suggestions for optimizing

In preliminary design stages, it is rarely required to trace all of the rays for all of the wavelengths at each field position during optimization. For this reason, execution times may be substantially decreased by limiting the number of fields and wavelengths used during optimization. If the weight of selected fields and wavelengths is set to zero, then the default merit function algorithm will skip the zero weighted fields or wavelengths when constructing the merit function. This results in fewer rays being traced, speeding execution.

For example, if the lens is being evaluated at five field points, it is possible that only the first, third, and fifth field need be included in the merit function. Of course, later in the design process all fields may need to be included and the default merit function reconstructed.

There are a few other tricks to improve performance. Avoid setting boundary operands on variables unless the optimized solution persists on implausible designs. Boundary operands add computational overhead. Use solves instead of explicit operands whenever possible. For example, use a curvature solve to control the focal length rather than an operand if possible.

Optimization is inseparable from the art of modern lens design, and only practice will make a designer a proficient user of optimization algorithms. Users who are expert at other software optimization algorithms will probably find Zemax easier to use, and with a little practice, the mechanics of using the interface will slip into the subconscious, and the designer can concentrate upon the design itself. If you are new to computerized optimization of lenses, there is no better way to learn than to practice.

The global optimum

The design that yields the lowest possible value of the merit function is called the "global" optimum and is by definition the best possible design. However, there is no known optimization algorithm that can universally find the global

optimum for an arbitrary design problem, unless you consider "direct search" an optimization algorithm (in other words, try all of the infinite number of possible solutions to see which is best). The art of optical design with computer assistance has two basic components. First, the designer must be able to determine a suitable starting design, and second, he or she must play the role of supervisor during the optimization process. A good supervisor knows when and how to back up and coax the program into a more fruitful direction.

Unfortunately, this often requires considerable experience, and even more often, excessive tedium. An experienced designer uses a combination of intuition, analysis, and luck in searching for new, better, design forms. Zemax provides an automated capability for performing this global optimum search; the feature is described the "Global Optimization" section of "The Optimize Tab" help file.

Sequential Optimization

The following sections are considerations for optimization in Sequential UI Mode.

Pitfalls with the default merit function

To set up the default merit function, see the "Optimization Wizard" section of "The Optimize Tab"

The default merit function is easy to set up (with the Optimization Wizard), numerically efficient, and suitable for a large number of optimization problems. However, most optical designs require extensions or modifications to the default as the design progresses. Zemax offers significant flexibility in the definition of the merit function, as described in the following sections.

Note that if the field or wavelength values or weights are changed, you must reconstruct the default merit function. If you are using the RA algorithm, reconstruct the default merit function if the vignetting influence changes appreciably during optimization.

If field or wavelength values or weights are changed, reconstruct the merit function.

Optimization with apodized beams

If no pupil apodization has been specified (see "Apodization Type" in the System Explorer) then Zemax assumes uniform illumination when constructing the default merit function. If the illumination is not uniform, then the rays in the default merit function are weighted according to the apodization factor. Since the rays selected may be insufficient to adequately represent an apodized beam, use a larger number of rays (described previously) when using apodization factors.

Using MTF operands

The diffraction based MTF operands (MSWT, MSWS, MSWA, MTFT, MTFS, MTFA, MTHT, MTHS, and MTHA) and the geometric MTF operands GMTA, GMTS, and GMTT provide a capability to directly optimize the MTF (see the list of operands in Optimization Overview). This is a powerful capability, however, using MTF operands requires some care.

The MTF cannot be computed accurately if the aberrations are too large. Any optical system for which the MTF plot produces invalid data will also produce meaningless data during optimization. Large aberrations also tend to produce MTF curves that drop to zero at some spatial frequencies, and then the MTF increases again at higher spatial frequencies. Local optimization of the MTF at frequencies beyond the first zero in the MTF curve is usually ineffective because the MTF must actually decrease before increasing as the aberrations improve, and the local optimizer is always looking for incremental improvement. The MTF operands should not be used if the starting MTF is beyond the first zero in the MTF curve at the desired spatial frequency. If the aberrations are large, MTF optimization is likely not required nor useful. An alternate approach is to first optimize using RMS wavefront error. Systems with low RMS wavefront error will have reasonable MTF performance. After the design is very nearly in the final form, then MTF optimization may be used for a final "touch-up".

MTF optimization is slower than RMS spot radius or RMS wavefront error optimization. Diffraction MTF is slower than Geometric MTF; however, the Geometric MTF is an approximation that is only reasonably valid for systems with circular or elliptical pupils and little or no apodization. Huygens based MTF can be many orders of magnitude slower than RMS optimization, and is thus not recommended for optimization. The slow execution speed may be noticeable when updating the Merit Function Editor display, and entering and exiting the optimization dialog box.

Note that if you use both MTFT and MTFS (or GMTT and GMTS) for the same field and wavelength data, they should be placed on adjacent lines in the editor; otherwise, the MTF is computed twice. If the sampling is too low, or the aberrations too large for accurate computation of the MTF, then the MTF operands return zero rather than a meaningless number.

Optimizing glass selection

Optimization of materials is handled somewhat differently than other data. Optimizing the glass choice directly is a difficult and unpredictable process because there does not exist a continuum of glasses on the glass map. There are two methods for dealing with this problem: by using model glasses (described below) or by using glass substitution (see "Using Glass Substitution" in the Global Optimization section). Glass substitution is usually the superior method.

Optimizing using model glasses

The model glass method is to idealize the glass dispersion using a few numerical parameters, and then optimize these parameters while constraining either the parameter values or the computed index values to be similar to available glasses.

To use the model glass method, first change the glass of the appropriate surface to a "Model" solve type using Lens Data Editor. See "Material Solves" for information on this setting in Lens Data Editor section of "The Setup Tab" help file. For information on model glasses, see the "Using model glasses" section of the "Using Glass Catalogs" reference file. With the "Model" solve type, Zemax will make a suitable guess for the index, Abbe number, and partial dispersion; you only need to change the values if you wish.

As stated in the "Material Solves" section referenced above, unconstrained glass optimization usually will lead to very high index materials being selected. This is because surfaces with high refractivity (a large difference in index across the boundary) need less curvature than low refractivity surfaces to have the same optical power. Lower curvature surfaces introduce less aberration.

Unfortunately, high index materials are expensive, heavy, harder to fabricate, and may be brittle, delicate, or susceptible to stains and scratches. Also, very high index materials do not always exist; there are few glasses (for the visible spectrum) available with an index higher than about 1.9. The Abbe number also is limited to the range of roughly 20 to 80. Therefore, it is essential to limit the index (Nd) and Abbe (Vd) numbers to reasonable ranges during optimization. The partial dispersion deviation ($\Delta P_{g,F}$) also must be limited in range.

There are two ways to limit the Nd, Vd, and $\Delta P_{g,F}$ values. The simplest way is to add the RGLA operand somewhere in the operand list. The RGLA operand measures the "distance" on the glass map between the index, Abbe number, and partial dispersion of the model glass to the closest glass in the currently loaded catalogs. For example, if you are optimizing the index and Abbe and you have specified that the Schott and Hoya catalogs are used (specified in the System Explorer), the RGLA operand computes the "distance" to each glass in these catalogs. If the smallest "distance" is less than the target value specified for the RGLA operand, then the boundary condition is met, and the value of the operand is equal to the target. If the closest glass is farther than the target value, then the RGLA value is the actual "distance". The "distance" is defined by the square root of the weighted sum of the squares of the difference between the index, Abbe, and partial dispersion terms for two glasses. The "distance" between any two glasses is given by

$$ d = [W_n(Nd_1 - Nd_2)^2 + W_a(Vd_1 - Vd_2)^2 + W_p(\Delta P_{g,F1} - \Delta P_{g,F2})^2]^{\frac{1}{2}}, $$

where the factors Wn, Wa, and Wp weight the various terms. The weighting factors may be user defined on the RGLA operand parameter list, or if left at zero, will default to 1.0, 1E-04, and 1E+02, respectively.

The best way to use RGLA is to specify the surface range that covers all of the surfaces you are optimizing. For a target value, start off with 0.05. This will allow the glasses to easily move all over the glass map, since the spacing between various glasses is usually less than 0.05. After optimization, decrease the target to roughly 0.02 and reoptimize. This will encourage the optimized system to choose index and Abbe numbers reasonably near actual glasses.

The other method for constraining index and Abbe values is to use the MNIN, MXIN, MNAB, and MXAB controls. These operands are mnemonics for Minimum and Maximum Index and Abbe values, and they are documented in the tables in "Optimization Operands". These operands can be used to restrict the optimization to specific rectangles on the glass map. It may be useful to use RGLA along with MXIN, for example, to restrict the glass selection to existing glasses with an index lower than some value.

At some point you will want to convert your variable index data back to a real glass. To convert from a model glass to the closest real glass, remove the model glass status using Ctlrl-Z, or choose the glass solve type as "Fixed". There will usually not be a perfect match between the optimized Nd and Abbe values and those of an actual glass in the current catalog. However, Zemax will search through the catalog and find the "best fit" glass using a least-squares criterion similar to the RGLA definition above (the partial dispersion term is omitted). If a glass substitution template is being used, then the template will also be considered, and only glasses meeting the template specifications will be considered. For information on templates, see "Glass Substitution Template" in "The Optimization Tab" help file. The glass in the catalog which differs the least from the variable index parameters is the glass selected. This glass is also reported on the "Surface Data" feature (in the Analyze Tab, select Reports, Surface Data). The index of refraction data shown is that calculated from the Nd and Abbe values, not the best fit glass. After converting from a model glass to a real glass, another optimization run is generally required. For systems with delicate chromatic aberration balancing, the best glass choice might never be found using variable glasses, because the model glass dispersion is never identical to the dispersion of a real glass.

Optimizing extra data

This discussion is only relevant for users of Zemax-Professional and Zemax-Premium.

Certain surface types supported by Zemax-Professional and Zemax-Premium, such as the Zernike, Zernike phase, extended polynomial, and binary optic surface use the extra data values. These extra data values can be edited, loaded from text files, and may be used as variables for optimization in the Lens Data Editor.

To make an extra data value a variable, see the Lens Data Editor "Solve Types" section of "The Setup Tab" help file.

There are also several boundary constraints for use with the extra data values. XDVA, XDGT, and XDLT are extra data value, greater than, or less than, respectively. The Surf value indicates the surface number to which the operand applies, and Edv# is used to specify which of the extra data values is to be used.

Optimizing objects in a non-sequential group with sequential rays

Optimizing variables within a non-sequential group is fundamentally no different from optimizing other numerical parameters. Variables are set in the same way as for parameters in the Lens Data Editor. However, optimizing non-sequential object properties is difficult because of the unpredictable way in which rays may (or may not) propagate through a non-sequential group. For non-sequential objects such as prisms, usually a small change in position or size of the prism does not dramatically affect the ray path. However, for objects such as light pipes, a small change in the object definition can dramatically affect the ray path. Rays that once propagated through an object may miss the object completely if the object position or angle changes slightly. This usually causes severe errors in the computation of derivatives, and the optimization performs either poorly or not at all.

Another problem with some non-sequential systems is the exit pupil may not be a reasonable image of the entrance pupil. For this reason, Rectangular Array rather than Gaussian Quadrature should be used if the system is a non-imaging system that does not form an image of the entrance pupil at the exit pupil.

For these systems, optimization may proceed more effectively using the global optimization algorithms, which do not rely exclusively on derivative computation.

Optimizing with the IMAE operand

The IMAE operand estimates the efficiency of an optical system by launching many rays into the entrance pupil; computing the fraction of rays that pass through all surface apertures to any surface. Optimization with this operand may not proceed smoothly if only hard-edged surface apertures, such as the circular aperture, are used. This is because Zemax estimates the derivative of operand values by making very small differential changes in the value of each variable, then computes a finite difference of the operand value. For the IMAE operand, a small change in the value of a variable may not change the efficiency estimate; if no rays are close enough to an aperture to change from being vignetted to unvignetted or vice-a-versa.

The solution is to replace hard edged apertures with soft-edged apertures placed on a user-defined surface. A soft-edged aperture has a transmission that is unity over most of the clear aperture, but near the edge the transmission drops to zero gradually over a small region, rather than abruptly.

Filter functions for doing this are included with Zemax as sample DLL files; see the "User Defined Surface" section of the "Sequential Surfaces" reference file for details. See in particular the discussion of the US_FILT4.DLL sample.

IMAE uses the current saved settings for the image analysis feature; except for "show" which is always set to spot diagram for this computation. See the "Geometric Image Analysis" section of "The Analyze Tab" help file.

Using gradient index operands

There are several optimization operands which are used to control the properties of gradient index materials during optimization. Some of them are described below.

DLTN

DLTN is used to control the maximum total change in index within a gradient index lens. **Surf** defines the surface number, and **Wave** defines the wavelength number. DLTN is defined as:

$$DLTN = n_{max} - n_{min}.$$

The min and max index values are computed at the extreme z coordinates, Z min and Z max. Z min and Z max are the Z coordinates of the minimum and maximum axial positions of the blank used to make the lens, before the shaping begins. For a convex surface, they correspond to the vertex. For a concave surface, they correspond to the maximum sag at that surface.

LPTD

LPTD is used to control the profile of the gradient within the material. **Surf** is used to define the surface number of the gradient index surface. LPTD is an acronym for LightPath Technology Delta, and the constraint is used to keep a nonlinear profile monotonically increasing or decreasing. It only needs to be used when the quadratic or cubic term of the axial gradient is variable. This operand only affects GRIN 5 surface types.

The LPTD operand should be used with a target of 0. The boundary constraint enforces the following conditions:

$$\frac{\partial n}{\partial z_{min}} > 0 \text{ and } \frac{\partial n}{\partial z_{max}} > 0, \text{ or } \frac{\partial n}{\partial z_{min}} < 0 \text{ and } \frac{\partial n}{\partial z_{max}} < 0.$$

Z min and Z max are the Z coordinates of the minimum and maximum axial positions of the blank used to make the lens, before the shaping begins. For a convex surface, they correspond to the vertex. For a concave surface, they correspond to the maximum sag at that surface. If the residual value of the operand is less than zero, then the target may be decreased slightly (try 0.1). Changing the target is usually more effective than increasing the weight. The value of the LPTD operand must be zero for the blank to be fabricated. Always check the gradient profile to make sure the slope does not change sign.

Global Optimization

The conventional means of optimizing lenses has for decades been the use of the damped least squares algorithm (DLS). DLS has many attractive features; it is efficient, and it is very good at finding the "local" minimum of the merit function. In this context, the word local means the lowest value of the merit function that can be reached from the current position in solution space without ever increasing the merit function (this is something of an idealization, in reality DLS can hop over small regions of increased merit function).

To visualize this problem, imagine you are on a hike, and are trying to find the bottom of a valley from a starting point on the side of a hill. You would like to find the lowest point in the valley. Suppose you cannot see the valley; it is very foggy and all you can see is the terrain very close to where you now stand. You can determine which way is downhill, and proceed in that direction until the slope becomes uphill again; at which point you find a new downhill direction. You might repeat this procedure over and over until a point is reached where all directions are uphill. This lowest point is then a local minimum, and the bottom of at least this valley.

The problem with this method is that once you have arrived at the local minimum, there is no known way (for the general optimization problem) to determine if there is not a better, lower minimum somewhere else. For example, if you were to walk uphill in any direction from this point until you reached a local peak, and then were to proceed onward downhill into the next valley, you will eventually reach a new local minimum. Is this new minimum lower or higher than the previous valley? The only way to find out is to take the hike!

You might ask, since computers are so fast, why not just try out every possible configuration to see which is best? To get a feel for the scope of the problem, consider a cemented doublet lens with six degrees of freedom (degrees of freedom manifest themselves as variables for optimization). If you assume that each variable can take on 100 possible values (a coarse sampling), then there are 1E+12 different possible systems. If each system evaluation requires the tracing of 20 rays (a low estimate) and you can trace 1,000,000 rays per surface per second, then the time required is about 8E+07 seconds, or about 2.5 years. For a four-element lens (16 variables) evaluated at three fields and three wavelengths, using 100 rays for evaluation would require 1E+32 system evaluations or many billions of times the age of the universe.

There are approaches to the global optimization problem that (thankfully) do not require unreasonable computational effort. These algorithms include simulated annealing, multistart, expert systems, neural networks, and others. All of these algorithms have strengths and weaknesses which are beyond the scope of this chapter.

Global optimization capabilities of Zemax

There are two separate global optimization algorithms in Zemax, each with a different purpose. The first algorithm is called simply "Global Optimization", and it is used to find new design forms given only the merit function and a starting design. See the "Global Optimizer" section of "The Optimize Tab" help file. Global Optimization uses a combination of genetic algorithms, multi-start, conventional damped least squares, and some expert system heuristics to search for new design forms. The Global Optimization algorithm is very good at finding promising design forms, however, it does not usually produce "finished" designs. The second algorithm is used for this purpose.

The second algorithm is called "Hammer Optimization" (lens designers often talk about hammering on a design to squeeze out the last bit of performance). See the "Hammer Optimizer" section of "The Optimize Tab" help file. The Hammer Optimization algorithm is used for exhaustively searching for the optimum solution once a reasonably good starting point is found, presumably by prior experience or the Global Optimization algorithm. The Hammer Optimization algorithm only requires a partially optimized lens and merit function in the form of a ZMX file.

Although the global optimization algorithms are extremely useful, it is important to realize that there is no guarantee that the true global optimum will always, or even occasionally be found. Of course, there is no way to even determine if any solution is the global optimum, even if it is the best you have ever found (remember the universe-lifetimes scale of the problem).

It is important to realize that there is no guarantee that the true global optimum will always, or even occasionally be found.

Both the Global and Hammer Optimization algorithms require extensive computational effort to be effective. These algorithms are not intended to be used interactively - that is what Local Optimization is for. If you set up global optimization, and watch while the computer works, you are bound to be disappointed. Global Optimization is highly effective when you set the problem up and let the computer run for many hours, or even several days, but not for ten

minutes. The ideal situation is to set up the problem before stopping work for the night, and let Global Optimization (or Hammer Optimization, depending upon your requirements) work overnight. In the morning, you should have useful results to look at.

The Global Optimization algorithm

Before you begin the Global Optimization, you must come up with a very rough starting point. "Very rough" means the design has the correct number of surfaces, a defined stop surface, and initial glasses selected. The fields and wavelengths must be defined. You also need to define a merit function; see the "Merit Function Editor" and the "Optimization Wizard" sections of "The Optimize Tab" help file for details on this procedure.

The very rough design can be parallel plates of glass with a curvature solve on the last surface to control the focal length. If a solve is not used to control the focal length, then the system should have at least the approximate focal length desired. Also, the variable parameters must all be defined. The lens must be saved before initiating Global Optimization. Zemax uses the starting focal length as a scaling parameter, so the initial design should have at least the approximately correct focal length!

Global Optimization will rarely find the global optimum by itself. The reason is that the Global Optimization concentrates the effort on finding new, promising design forms rather than converging exactly on the best possible solution for each form. This latter job is left to a separate algorithm, called "Hammer Optimization", described in the following section.

The Hammer Optimization algorithm

After reviewing the design forms generated by Global Optimization, you will probably want to investigate one or two of them. The one with the lowest merit function is not always the best (although it should be if you designed your merit function well). For example, the second-best solution may be easier to fabricate. Whatever criterion you use to determine the most promising solution, you now want to find the best design possible using this selected lens as a starting point.

The Hammer Optimization algorithm can also be used effectively on partially optimized designs that were not generated with Global Optimization.

For more information see the "Hammer Optimizer" section of "The Optimization Tab" help file.

Global optimization of glass selection

If a glass is made into a "model" glass described by the index, Abbe, and partial dispersion deviation, then the parameters to the model may be made variable and optimized just like any other numerical parameters. However, the model glass method has one serious drawback. After a good solution is found using model glasses, a conversion must be made back from the model glass to a real glass. The design must then be reoptimized using the new glass selection.

Unfortunately, for many systems the newly optimized design will perform worse than the model glass design. Even more frustrating is that the optimal design using real glasses may have a different form than that found using the model glass.

Traditionally, to find a better combination of glass types, the designer would locate alternate glasses on the glass map, substitute in the new selections, and reoptimize. If the new solution was better, the glass choices would be retained; otherwise, a new set of glasses would be evaluated. The procedure would continue as long as the designer could bear to continue the search.

Using glass substitution

The glass substitution method is to directly alter the glass types, and then reoptimize to see if the new glasses yield a better solution. This method can be used manually, by changing the glass type and then reoptimizing, or the process can be automated by allowing glasses to have a "substitute" status associated with them. If a glass is marked (using the glass solve dialog box) as a substitute glass, then the global optimization algorithms (both Hammer and Global Optimization) automatically perform iterative substitution of similar glass types during optimization. This allows Zemax to optimize not only the numerical prescription values such as radius and thickness, but also allows direct optimization of real glass selection, without resorting to idealizing the glass dispersion.

To use the substitution feature, set the status of each glass free to change to "substitute" on the glass solve dialog box (double clicking on any glass name is the quickest way to reach this dialog box).

Once the glass substitution status is defined, then invoke either the Hammer or Global Optimization optimizers. Zemax will automatically alter the glass types during the search for better designs. By default, Zemax will choose any glass from any of the currently active catalogs, which are specified in the System Explorer. Glass substitution does not use table glasses or MIL number glasses.

See "Using glass substitution" in the "Using Glass Catalogs" reference file for a discussion on this method for optimizing glass selection.

Restricting selected glasses

In practice, it is usually necessary to restrict the available glasses for substitution for these reasons:

Not all "glasses" in the catalogs are actually glasses; some are liquids, gases, crystals, or plastics that may not be acceptable for use.

Many glasses are very expensive, heavy, brittle, or have other undesirable mechanical properties.

Perhaps only certain catalogs are desired for the newly chosen glasses; while other catalogs must still be used for other surfaces in the optical system.

For these reasons, Zemax provides several optional methods to restrict the pool of available glasses used for substitution:

By defining a specific catalog only be used for any given surface.

By defining a "template" which places limits on the cost, acid resistance, stain resistance, and other properties of substituted glasses.

By "excluding" glasses on a case-by-case basis in the glass catalog.

By defining "penalties" in the merit function which discourage selection of glasses with undesirable properties. The glass solve dialog box permits specification of a catalog name to use for candidate substituted glasses.

If no catalog name is given (i.e. the catalog field is blank) then glasses may be selected from all catalogs selected for use in the System Explorer. If a catalog name is given (i.e. Hoya) then only glasses from that one catalog will be selected. This allows different surfaces to select from different catalogs, if required. It is also useful for restricting the number of glasses considered for substitution in a very general way. Note that user defined catalogs may be specified. No extension should be specified, Zemax automatically appends the .AGF extension.

A Glass Substitution Template may be defined to restrict which glasses from all catalogs are chosen for substitution based upon their cost, AR, SR, FR, CR, and PR code values. See the "Glass Substitution Template" section in "The Optimize Tab" help file for information on defining a template.

To prevent only certain glasses from being chosen during optimization, choose "Exclude Substitution" for the glass that should be avoided on the glass catalog dialog box; see "Description of catalog data" in the "Using Glass Catalogs" reference file. The advantage in excluding glasses is that other glasses in the catalog may still be used, without having to define a separate catalog.

Penalties are operands in the merit function such as GCOS, GTCE, and INDX which are used to increase the merit function if glasses have unacceptable properties. This is the least efficient method, because glasses may still be selected, and the subsequent lens optimized, even if the resulting merit function ends up being too high to be considered a good solution. However, it is useful for defining relationships between glasses, such as minimizing the difference of GTCE between two glasses which form a cemented surface (to prevent the lens from breaking under thermal stress).

Suggestions for using global optimizers

There are several techniques for maximizing the performance of the global optimization algorithms (both Global Optimization and Hammer):

1) If possible, place the stop on the first surface. If your entrance pupil is embedded in the system, you can model this by using a dummy first surface for the stop and then use a negative thickness to get to surface 2. This enhances performance because Zemax does not need to calculate where the entrance pupil is. You can make this thickness variable if appropriate. This technique does not work well for systems which inherently have entrance pupil distortion, such as wide angle lenses.

2) Use a marginal ray angle solve on the last glass surface curvature to control the effective focal length instead of the EFFL operand. Dropping one variable decreases the dimensionality of the problem significantly, and the EFFL operator does require one additional ray trace, slowing down the merit function evaluation.

3) Use a marginal ray height solve (for zero ray height) on the last thickness before the image surface. Most lenses are well corrected for the 0.7 pupil zone on axis. You may want to use another pupil zone if your intuition guides you to. This solve will ensure that every design generated by Global Optimization is in focus, just as the curvature solve ensures the correct focal length. These two solves together can improve the performance of the Global Optimization algorithm by several orders of magnitude. For Hammer optimization, replace the solve with a variable to allow for optimal defocus.

4) Use the MNCT and MNET operands. These are essential for avoiding negative center and edge thicknesses. Zemax uses these boundary constraints to determine the appropriate range of each variable in the search. The solutions will wander into these unacceptable regions of solution space unless you specifically forbid it.

5) Keep your merit function as simple as possible. Using 2 or 3 rings in the default merit function construction (see the chapter "Optimization") is often a good idea. You can always go to more rings for the Hammer optimization.

6) Be willing to run several long runs; perhaps one before starting a design problem to get ideas for design forms, a few more to explore various promising designs, and a last run to ensure the design you have settled on cannot be improved (Hammer is particularly good at this last task).

7) Use substitute glasses rather than model glasses, especially when using Hammer. There is not much point in using Hammer on a design that uses model glasses, because ultimately the glasses will need to be converted back to real glasses and then optimized again. Substitute glasses are highly useful, especially late in the design process when the design form has been determined.

8) The Hammer algorithm may be terminated and then restarted without significant loss of information. As the algorithm proceeds, it automatically saves any improved designs.

9) If a Global Optimization is terminated, the search may be resumed at a later time by choosing resume instead of start.

10) Some computes may be programmed to go into "sleep" mode if there is no user activity for a set amount of time. This feature may need to be disabled to allow Zemax to continue running, even if there is no input from the keyboard or mouse for long periods.

Summary

Finally, it is again stressed that the global optimum search feature may not ever find the global optimum to any particular problem. It is very good at finding alternative design forms that would have been tedious to discover by hand. It is a very powerful tool to add to your toolbox; it is not a substitute for optical design skills.

The Global Optimization algorithm does have a strong random component, and therefore no two runs will yield the exact same solutions every time. Sometimes the solutions will be worse, sometimes better, but usually they are just different for runs of similar duration.

Thermal Analysis of Optical Systems

Environmental factors such as temperature and pressure can affect the performance of optical systems. There are three primary factors to consider. First, the index of refraction of glass depends upon both temperature and wavelength; relative indices which are measured with respect to air also change with pressure. Second, glass expands and contracts with temperature, which can change the radius, thickness, or other dimensions of a lens. Third, the spacings between lenses changes due to the expansion and contraction of the mounting material.

The thermal analysis features of Zemax can account for all these effects. By accounting for thermal effects, Zemax can be used to analyze and optimize a design for any specific temperature or for a range of temperatures.

The index of refraction data given by the dispersion formulas are referenced to a specific temperature and pressure, which is typically 20 or 25 degrees Celsius (depending upon the manufacturer) and 1 atmosphere. Also, index of refraction data is by convention referenced to that of air, which means thereafter the air has an index of unity. The index referenced to air is called the relative index of refraction. When the index is referenced to vacuum (which truly has an index of unity) then the index is called the absolute index of refraction. The difference between these two references for any glass is a function of wavelength, temperature, and pressure.

Defining temperature and pressure

Two user provided values define the environment: the system temperature in degrees Celsius and the air pressure measured in atmospheres. These values are set on the dialog box which can be found under the Environment section of the System Explorer. By default, any defined temperature and pressure applies to all surfaces in the optical system. However, optical systems which require multiple temperatures to be defined in the same system are also supported. This is required for optical systems that have some lenses in a different pressure (such as a vacuum) or in a region that is cooler or warmer than the system environment.

Defining wavelengths

Wavelength data are always measured in micrometers referenced to "air" at the system temperature and pressure. The default system temperature is 20 degrees Celsius, and the default air pressure is 1.0 atmospheres. If the system temperature and/or pressure is modified, or under the control of multi-configuration operands, care must be taken to adjust the wavelengths to the new air temperature and pressure.

Wavelength data is entered in the "Wavelength Data" section of the System Explorer; see the "Wavelengths" for details.

Index of refraction computation

Zemax always uses relative, not absolute index data. One way to think about relative index is that the index of "air" (indicated by a blank entry in the glass column) is always defined to be exactly 1.0 at all wavelengths at the system temperature and pressure. Air and glass spaces at other temperatures and pressures are normalized relative to 1.0.

There are several steps required for Zemax to compute the relative index data. The basic steps for computing the index for each glass type at each wavelength are:

-Scale the wavelength to air at the reference temperature of the glass and a pressure of 1.0 atmospheres.

-Compute the relative index of the glass at the reference temperature from the dispersion formula.

-Compute the index of air at the reference temperature of the glass.

-Compute the absolute index of the glass (relative to vacuum) at the reference temperature of the glass.

-Compute the change in absolute index of refraction of the glass at the surface temperature.

-Compute the index of air at the system temperature and pressure

-Compute the index of the glass relative to the air at the system temperature and pressure.

The end result, which is the index of the glass at the surface temperature and pressure relative to air at the system temperature and pressure, is what is used by Zemax for ray tracing. The index of a surface made of air is treated in the same way. This means that air surfaces at different temperatures and pressures from the system temperature and pressure will have slightly lower or higher index. For example, if the system pressure is 1.0 and the pressure of one air surface is set to 0.0, the index of this one surface will be approximately 0.99973. If the system pressure is 0.0 and the air space has a pressure of 1.0, the index for that surface will be approximately 1.00027. Remember, air at the system temperature and pressure is defined to be 1.0, all other indices are relative.

Note also that Zemax can easily model systems used in a vacuum by changing the air pressure to zero. If only some surfaces are at vacuum, this can be set up using the TEMP and PRES commands on the multi-configuration editor.

For the index of refraction of the glass, Zemax uses the dispersion formulas and data stored in the glass catalog. For details, see "Description of catalog data". For the index of air, Zemax uses the following formula:

$$n_{air} = 1 + \frac{(n_{ref} - 1)P}{1.0 + (T - 15) \cdot (3.4785 \times 10^{-3})}$$

where

$$n_{ref} = 1 + \left[6432.8 + \frac{2949810\lambda^2}{146\lambda^2 - 1} + \frac{25540\lambda^2}{41\lambda^2 - 1} \right] 1.0 \times 10^{-8},$$

T is the temperature in Celsius, P is the relative air pressure (dimensionless) and is measured in micrometers. This formula for the index of air is from F. Kohlrausch, Praktische Physik, 1968, Vol 1, page 408. The change in absolute index of the glass with temperature is given by the following expression:

$$\Delta n_{abs} = \frac{n^2 - 1}{2n} \left[D_0 \Delta T + D_1 \Delta T^2 + D_2 \Delta T^3 + \frac{E_0 \Delta T + E_1 \Delta T^2}{\lambda^2 - S_{tk}\lambda_{tk}^2} \right],$$

where n is the relative index at the reference temperature of the glass; ΔT is the change in temperature relative to the reference temperature of the glass (ΔT is positive if the temperature is greater than the reference temperature of the glass); D_0, D_1, D_2, E_0, E_1, and λ_{tk} are constants that are provided by the glass manufacturer to describe the thermal behavior of the glass; and S_{tk} is the sign of the λ_{tk} constant (i.e. $S_{tk} = 1$ if λ_{tk} is positive and $S_{tk} = -1$ if λ_{tk} is negative). This model for the change in the glass index was developed by Schott Glass Technologies, Inc.

The six constants must all be provided in the glass catalog for the computation to be valid. The default values of zero for all six constants yields zero for the change in index; therefore if no thermal data has been added to the catalog, no thermal effects are considered. Zemax cannot compute thermal index variation for any glass type without these six constants provided. However, some approximations are available if the six coefficients are not available. See the section "Adding thermal index variation data" for details.

Environmental effects on index for gradient index, MIL number, and model glasses

Zemax only partially considers environmental effects on the index of refraction of gradient index surfaces, MIL number glasses, or model glasses which are described only by index and Abbe number. The relative index of refraction computed using these methods is adjusted to account for the change in the index of surrounding air at the system temperature and pressure. However, the change in the absolute index of these glasses with temperature and pressure is not considered, because no "dn/dt" data is defined for these glass types. However, it is possible to manually define the variations in properties across multiple environments using the multi- configuration editor.

Defining multiple temperature and pressure values

To analyze or optimize an entire lens at a specific temperature and pressure, all that is required is to define the relevant data on the environment tab of the general data dialog box described in "Environment". All the radii and thickness data are then assumed to be measured at that temperature, and Zemax computes the index data accordingly. However, the real power of the thermal analysis feature comes into play when a lens must be analyzed at or optimized for multiple environments, such as a broad temperature range or varying altitudes (or both). There are several new issues these lens systems introduce:

A means of specifying the nominal temperature at which radii and thickness are measured must be provided.

The change in index, radii, and thickness must be accounted for as the environment changes.

The thermal effects on mounting material must be considered.

Some surfaces may be at one temperature or pressure, while other surfaces are at a different temperature or pressure.

Zemax is up to the task in all particulars. The basic approach to setting up a multiple environment lens is:

Define the lens at some nominal temperature and pressure. This should be the environment under which the lens will be fabricated. All radii and thickness data will be specified only at this temperature and pressure.

Now define additional configurations using the multi-configuration feature (see "Multi-configurations"). In each additional configuration, the temperature and pressure will be specified, and special solves called "thermal pickup solves" will be used to adjust the radii and thickness data for each configuration. The multi-configuration operands that control temperature and pressure are TEMP and PRES.

TCE stands for thermal coefficient of expansion. When a glass element changes temperature, the linear change in size is given by the expression $L' = L(1 + \alpha\Delta T)$, where L is a linear dimension, α is the TCE, and ΔT is the change in temperature. As the material expands, the radius of curvature also expands. Therefore, both the thickness and the radii of a glass surface change linearly with temperature. The assumption of linearity is only an approximation, but a reasonably good one for most materials and temperature ranges.

The TCE coefficient is defined along with the glass dispersion data in the glass catalog. See "Using Glass Catalogs" for details on the TCE data.

Defining which parameters consider thermal effects

On the multi-configuration editor, there is a special solve called a thermal pickup. This pickup solve is used to compute a new value for a multi-configuration parameter based up on the temperature and pressure of the new configuration as compared to the "reference" configuration. Thermal pickup solves only affect data for certain types of multi-configuration values, as described below.

Radius of curvature (CRVT) values

If the multi-configuration operand is a CRVT, then the curvature of that surface for that configuration is computed from the curvature in the nominal configuration, the difference in temperature between the two configurations, and the TCE for that material.

If the glass type for a surface is specified in the catalog (such as BK7 or F2) then the TCE from the catalog is used. If the glass type is air (a blank entry) or "MIRROR" then there are two possibilities: if the previous surface glass type is a catalog glass, then the TCE for that glass is used, otherwise, the TCE value entered in the TCE column will be used. These rules have one important ramification: if an element is a cemented doublet, the TCE for the first surface will be

that of the first glass, while the TCE for the second and third surfaces will be that of the second glass. Zemax ignores the "stress" induced in the common surface. This assumption may not be accurate for a large temperature range. Mirrors surrounded by air use the TCE from the TCE column, mirrors preceded by glass are assumed to be made of that same glass, and therefore use the TCE for the preceding glass.

Thickness (THIC) values

If the operand is a THIC, then there are two possibilities. If the surface is composed of a catalog glass, then the TCE for that glass is used. Otherwise, the TCE specified in the TCE column of the surface is used. The TCE column is used for entering user-defined TCE data for the material used to manufacture the mount. There is one other important point: if the surface is not composed of a catalog glass, then the thermal expansion is computed along a length of material which extends from the edge of the surface to the edge of the next surface. Since the material expands along the length of the edges, not at the center thickness, this is a more accurate computation.

For example, suppose two lenses were separated by aluminum with a center thickness spacing of 80 mm. If the sag of the first surface (the rear of the first lens) lens was -5 mm, and the sag of the second surface (the front face of the second lens) was 8 mm, the total edge thickness is then 93 mm. If the TCE for aluminum is 23.50 E-6, then under a temperature change of +20 degrees the edge thickness would change from 93 to 93.0437 mm. Ignoring the change in diameter of the spacer and the change in the sag of the two lens faces (Zemax actually accounts for both of these effects, but for simplicity they are ignored here), the center thickness would then change to 80.0437. Note that this is a different amount of expansion than if the computation had only considered the center thickness.

Because the expansion along the edge accounts for the change in radii of the adjacent surfaces, even a TCE of 0.0 will result in a change of thickness if the radii change. To turn off thermal expansion of a thickness, do not use a TCE of 0.0, rather, delete the thermal pickup solve entirely.

Thermal parameter values

The thermal pickup solve properties for parameter values depends upon the parameter number and surface type. If the parameter is interpreted as having units of length, then the appropriate scaling is performed just like the radius of curvature described earlier. If the units are in powers of length, such as length squared or inverse length, then appropriate scaling is also performed. Otherwise, the thermal pickup solve ignores thermal effects and picks up the value from the nominal configuration.

Thermal extra data values

The thermal pickup solve properties for extra data values depends upon the surface type. Generally, only the "normalization radius" needs to be scaled. This convenience is due to the fact that thermal expansion can be considered a scaling of length. Since most extra data surfaces use dimensionless coefficients, only the normalization radius needs to be scaled. This works effectively for binary optic and polynomial aspheric surfaces equally well. For extra data surface types which do not use a normalization radius, the thermal pickup solve ignores thermal effects and picks up the value from the nominal configuration.

Special surface types, such as polynomial aspheres, binary optics, holograms, and other surfaces that use parameter or extra data need to have operands and thermal pickup solves added manually to the multi- configuration editor; the automatic thermal setup tool does not add these operands automatically.

All other values

All other values are straight pickup solves, and the value will be identical to that in the nominal configuration. Thermal effects are ignored.

Defining multiple environments within a single configuration

Sometimes it is required to have different parts of the optical system at different temperatures and pressures. Note this is different from an entire system being at different environments across multiple configurations.

Groups of surfaces can be assigned their own temperature and pressure using the TEMP and PRES multi- configuration operands, even if there is only 1 configuration defined. The key is that each TEMP and PRES operand defines the environment for all operands that follow in the multi-configuration editor. The last TEMP and PRES operands listed in the editor define the "global" environment, which will apply to all data not listed in the multi-configuration editor.

For example, suppose an optical system model requires surfaces 1-5 to be at a temperature of 20 degrees C, while surfaces 6-10 are at 50 degrees C. The first operand listed should be TEMP (this same discussion applies to PRES), defining the initial environment of 50 degrees C. All the curvatures, thicknesses, semi-diameters, glasses, and other values for the surfaces 6-10 should be listed after the TEMP operand. Then the list should end with another TEMP, this one defining the 20 degree "global" environment. The resulting system will be evaluated at the respective temperatures (and/or pressures) for each surface.

It is extremely important to understand the two basic rules:

1. All multi-configuration editor operands that follow a TEMP or PRES operand are evaluated at that temperature or pressure.

2. The last TEMP and PRES listed in the multi-configuration editor define the temperature and pressure for all other lens data, on the multi-configuration editor or not.

By far the most important step in setting up a complicated multiple environment lens is to check the set up carefully. Two excellent tools for doing this are the index of refraction data and multi-configuration data table on the prescription report. This table lists the temperature and pressure for each glass type, and the thermal pickup relationships.

Checking the thermal pickup solves on each parameter is also a good idea; the data should be at least partially verified by hand to verify the correct temperature range and expansion are being used.

Automatic thermal setup

A tool for automatically setting up a lens for thermal analysis is described in "Make Thermal".

Adding TCE data

There are two kinds of TCE data. For surfaces which use a glass type named in one of the catalogs (such as BK7 from the Schott catalog), Zemax uses the TCE data specified in the catalog. See "Using Glass Catalogs" for the description of α, the thermal coefficient of expansion value.

If the surface does not use a catalog glass, then the TCE value is extracted directly from the TCE column in the Lens Data Editor. The TCE column is the last column in the spreadsheet, to the right of the parameter columns.

TCE is measured in units of 1E-06 per degree C. Therefore, a value of 23.50 E-06 per degree C would be entered as 23.5. Zemax automatically considers the 1E-6 factor when computing thermal effects.

Modeling gases and liquids

Once a material is defined in the glass catalog, Zemax uses the TCE specified for that material to determine the thermal expansion of the radius, center thickness, and other data for the surfaces using the material. However, if the material is not a solid, but is instead a gas or a liquid, then generally the thermal expansion is not governed by the material properties, but is instead determined by the edge thickness of the mounting material.

In this special case, Zemax needs to use the TCE supplied in the Lens Data Editor to define the mounting material properties, rather than the TCE supplied in the glass catalog. This can be accomplished by setting the "Ignore Thermal Expansion" switch for the material in the glass catalog.

Adding thermal index variation data

The variation of index of refraction with temperature, air pressure, and wavelength is modeled for any glass using the polynomial expression given earlier. The expression requires six coefficients to define the temperature and wavelength dependence on the change in absolute index. It is often the case that for materials added by the user, the six coefficients describing the variation are not available. However, most glass catalogs contain at least a single linear approximation of the ratio of index change with temperature change. This value is called dn/dt. If the only data available is a single dn/dt, then an approximation to the general expression can be made assuming all the coefficients except D_0 are zero:

$$\Delta n_{abs} = \frac{n^2 - 1}{2n}[D_0 \Delta T],$$

which implies that a reasonable approximation to D_0 is given by

$$D_0 = \frac{2n}{n^2 - 1}\frac{dn}{dt}.$$

The D_0 needs to be calculated and entered into the glass catalog. The relative index at the center wavelength measured at the reference temperature for the glass is an adequate value for the index n. Extreme care should be taken to ensure this is an adequate approximation by subsequently checking the computed index values at various wavelengths and temperatures. Note that the dn/dt data should be the absolute, not the relative dn/dt.

Using a single dn/dt value to estimate the D_0 term is only a rough approximation. The actual index variation with temperature is not likely to be linear over any extended wavelength or temperature range. Therefore, extreme caution and suspicion is warranted when using only a single dn/dt value.

Optimizing athermal lenses

To optimize an athermal lens, first define the multiple configurations required to simulate the lens at each temperature using the methods described in the previous sections. Then define the variables for optimization only in the nominal configuration. For example, suppose the nominal configuration is number 1, and configurations 2, 3, and 4 are defined using thermal pickup solves for every curvature and thickness. Make only the curvatures and thicknesses in the nominal configuration variable.

It is also possible to optimize the TCE of the spacing material between lens groups. To do so, set a variable on the value in the TCE column of the Lens Data Editor.

Limitations of thermal analysis

There are several limitations to the accuracy of the thermal analysis capabilities of Zemax. First of all, the TCE data should always be checked for accuracy over the temperature range of use. The index data coefficients should also be verified with the manufacturer of the glass being used.

The thermal analysis does not necessarily work correctly with tilted, decentered, or otherwise unconventional optical systems. The difficulty arises when computing edge thickness displacements on components that are not symmetric; for example, between two lenses tilted with respect to one another.

The thermal index and TCE data provided for Schott glasses comes from Schott, and they state the data is accurate over the temperature range from -40 to +80 degrees Celsius, and over the wavelength range 0.435 to 0.644 micrometers. The data may be extrapolated out to 1.06 micrometers with reduced accuracy. Data for other glasses is provided as is and the range of accuracy is unknown.

Because of the complexity of thermal effects modeling, none of the data should be trusted in critical applications and all computations, index values, and TCE data should be verified independently of Zemax. This is true even when working with Schott glasses within the ranges specified above.

Physical Optics Propagation

Introduction

This feature is only available in the Professional and Premium editions of Zemax. Physical optics propagation is complex, and the user is urged to read and understand this section before using the feature.

Geometrical optics is the modeling of optical systems by tracing rays. Rays are imaginary lines which represent normals to the surfaces of constant phase, called the wavefront. Either rays or wavefronts can be used to represent a beam. However, rays and wavefronts are propagated differently. Rays propagate along straight lines without interfering with one another, wavefronts propagate while coherently interfering with themselves. For this reason, the ray model and the wavefront model yield different representations of the beam as it propagates through free space or through optical components. The ray method is fast, flexible, and extremely useful for modeling almost any optical system. However, rays are not well suited to modeling certain important effects, primarily diffraction.

Zemax does have some ray based diffraction computations, such as the diffraction MTF or PSF. These diffraction computations make a simplifying approximation: that all the important diffraction effects occur going from the exit pupil to the image. This is sometimes called the "single step" approximation. Rays are used to propagate the beam from the object, through all the optics and intervening spaces, all the way to the exit pupil in image space. The ray distribution in the exit pupil, with transmitted amplitude and accumulated OPD used to compute the phase, is used to form a complex amplitude wavefront. Then, in a single step, a diffraction computation is used to propagate this complex amplitude wavefront to the region near focus.

Geometrical optics and the single step approximation work quite well for the majority of traditional optical designs, where the beam is not near focus anywhere except the final image. However, the model breaks down for several important cases:

When the beam comes to an intermediate focus, especially near optics that truncate the beam (rays by themselves do not predict the correct distribution near focus).

When the diffraction effects far from focus are of interest (rays remain uniform in amplitude and phase, wavefronts develop amplitude and phase structure).

When the propagation length is long and the beam is nearly collimated (collimated rays will remain collimated over any distance, real beams diffract and spread).

Physical optics is the modeling of optical systems by propagating wavefronts. The beam is represented by an array of discretely sampled points, analogous to the discrete sampling using rays for a geometric optics analysis. The entire array is then propagated through the free space between optical surfaces. At each optical surface, a transfer function is computed which transfers the beam from one side of the optical surface to the other. The physical optics model allows very detailed study of arbitrary coherent optical beams, including:

Gaussian or higher order multi-mode laser beams of any form (beams are user definable).

Beams may be propagated along any arbitrary field position (skew beams).

Amplitude, phase, and intensity may be computed at any surface in the optical system.

Effects of finite lens apertures, including spatial filtering, may be modeled.

Accurate computation of propagation through any optical component Zemax can model via ray tracing.

The physical optics model is generally more accurate at predicting the detailed amplitude and phase structure of the beam away from focus than conventional ray tracing. However there are some disadvantages to the physical optics propagation analysis:

> Physical optics is generally slower than geometrical optics.
> Because the entire beam array must be stored in computer memory at once, the required RAM may be quite large for large sampling arrays.
> The sampling limits the amount of aberration in the beam that can be accurately modeled. For highly aberrated systems, geometrical optics should be used.

The following sections will summarize the physical optics propagation algorithms and the information needed to properly use this advanced Zemax feature.

Support for multiple processors

The physical optics propagation routines in Zemax are designed to work on multiple CPU computers. Computers with two or more CPU's will execute the physical optics propagation feature faster than single CPU computers.

Diffraction propagation

The theory and methodology of diffraction propagation is well understood and widely detailed in available literature. The methods used in Zemax are based upon these references:

> Goodman, Joseph W., Introduction to Fourier Optics, McGraw-Hill, New York (1968).
> Lawrence, George N. "Optical Modeling", in Applied Optics and Optical Engineering, Volume 11, R. R. Shannon and J. C. Wyant, eds., Academic, New York (1992).

Only the material relevant to using the physical optics propagation feature in Zemax will be summarized here.

Representation of the electric field

The electric field may be represented in three dimensions as

$$\vec{E}(x, y, z) = E_x \hat{x} + E_y \hat{y} + E_z \hat{z}$$

where the E values are all complex and $\hat{x}$, $\hat{y}$, and $\hat{z}$ are the Cartesian unit vectors. The coordinate system used by Zemax is that the beam propagates primarily down a local Z axis. The Z axis used to represent the beam is aligned with a reference chief ray in each optical space, and therefore this Z axis is not generally the same as the Z axis defined by the Lens Data Editor which is used to position optics. Because the beam is propagating along the local Z direction, the first approximation made is to neglect the E_z component. Since the electric field must always be normal to the ray propagation direction, E_z can be reconstructed from other data when required, as will be described later.

By keeping track of the electric field components along both the X and Y axes, effects due to polarization may be studied, such as transmission and reflection losses, polarization aberrations, and of course the polarization state of the beam. If polarization effects are not required, the Y component of the field may be ignored, speeding the computations.

The Fresnel number

A very useful concept in physical optics modeling is the Fresnel number. Strictly speaking, the definition of the Fresnel number only applies to unaberrated rotationally symmetric beams with a finite extent. However, the concept is still useful in cases that do not meet these criteria. The Fresnel number depends upon the diameter of the beam, the radius of curvature of the wavefront phase, and the distance to an observation point where the complex amplitude of the field is desired. Conceptually the Fresnel number is the number of annular "Fresnel zones" from the center of the beam to the edge. Fresnel zones are the radial zones where the phase as seen from the observation point changes by π. A perfectly collimated beam will have a Fresnel number given by

$$F_n = \frac{2}{\lambda} [\sqrt{Z^2 + A^2} - Z]$$

which for Z greater than A reduces to approximately

$$F_n = \frac{A^2}{\lambda Z}$$

where A is the radial size of the beam and Z is the distance from the beam to the observation point. The Fresnel number becomes small as Z grows large.

For beams that are not collimated, the concept is the same. A converging beam will have a very small Fresnel number if the observation point is near focus. A perfectly spherical beam converging to focus will have a Fresnel number of zero, since there are no zones where the observed phase reaches π. As the observation point moves from the focal region, the Fresnel number increases.

Near and far field

If the Fresnel number is small, less than roughly 1, then the beam at the observation point is said to be in the "far field" relative to the current beam. For Fresnel numbers larger than 1, the beam at the observation point is said to be in the "near field" relative to the current beam.

It is important to consider the terms near and far as being relative to the propagation from the present location of the beam to the observation point at which the Fresnel number is computed, rather than having any rigid relationship to the beam position alone. For example, a beam in the exit pupil of an optical system is typically called the near field because the far field is at focus. However, a short propagation from focus to a slightly out of focus observation point is likely a near field propagation if the defocus is small.

The decision as to whether propagation is in the near or far field will determine the choice of diffraction propagation methods.

Angular spectrum propagation

It is well known that plane waves remain plane waves while propagating in homogeneous medium. A plane wave is represented by

$$e^{i\vec{k}\cdot\hat{z}}$$

where $\vec{k}$ is the wave vector, with magnitude $(2\pi)/\lambda$, and $\hat{z}$ is the local z direction. The vector $\vec{k}$ points along the normal to the wavefront in the direction of propagation. This normal vector has direction cosines α, β, and γ, where

$$\alpha^2 + \beta^2 + \gamma^2 = 1$$

The plane wave can then be written as

$$e^{\frac{i2\pi}{\lambda}(\alpha x + \beta y + \gamma z)}$$

Now recall the definition of the Fourier transform and the inverse Fourier transform:

$$FF[A] \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} [A]e^{-i2\pi(x\xi + y\eta)}dxdy \ ,$$

$$FF^{-1}[A] \equiv \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} [A]e^{i2\pi(x\xi + y\eta)}d\xi d\eta \ .$$

Let G = FF [E], that is, G is the Fourier spectrum representation of the electric field E . By definition then

$$E(x, y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} G(\xi, \eta)e^{i2\pi(x\xi + y\eta)}d\xi d\eta \ .$$

The electric field can therefore be interpreted as being the integral of a collection of plane waves propagating with direction cosines

$$\alpha = \lambda\xi, \ \beta = \lambda\eta, \text{ and } \gamma = \sqrt{1 - (\lambda\xi)^2 - (\lambda\eta)^2} .$$

Eliminating γ, and **making the approximation** that the plane wave propagates at a small angle with respect to the Z axis (for more information on these approximations search "Algorithm assumptions" in the help files), the plane wave equation can be rewritten

$$e^{\frac{i2\pi z}{\lambda}\sqrt{1-\alpha^2-\beta^2}} \approx e^{\frac{i2\pi z}{\lambda}} e^{\frac{-i\pi z(\alpha^2+\beta^2)}{\lambda}} .$$

The $e^{\frac{i2\pi z}{\lambda}}$ term is just a phase propagation term that is normally neglected. The term which depends upon α and β is the transfer function for a plane wave in free space. Defining $\rho^2 = \xi^2 + \eta^2$ the plane wave transfer function can then be rewritten as

$$G(\xi, \eta, z) = G(\xi, \eta, 0)e^{-i\pi\lambda z\rho^2} .$$

To propagate an electric field from one plane to another, the field needs to be Fourier transformed, the plane wave propagator applied, and then the resulting distribution inverse Fourier transformed. These operations may be summarized by defining the plane to plane (PTP) operator:

$$E(x, y, z_2) = PTP[E(x, y, z_1), (z_2 - z_1)] , \text{ where}$$

$$PTP(E, \Delta z) \equiv FF^{-1}[T(\Delta z)FF[E]] , \text{ and}$$

$$T(\Delta z) = e^{-i\pi\lambda\Delta z\rho^2} .$$

Note that the transfer function T(Δz) has unity amplitude but a complex phase. This phase varies slowly from point to point in the frequency domain representation G if $\lambda\Delta z\rho^2$ is small. But if $\lambda\Delta z\rho^2$ grows large, the phase variations become increasingly rapid. If the phase changes by more than about ⬜ ⬜ 2 between adjacent points in the finite array, the phase becomes ambiguous, and a phenomenon known as aliasing occurs. For this reason, the angular spectrum method works very well if the propagation distances are fairly short or if the beam is nearly collimated. Although the diffraction theory is accurate for any propagation distance, when the beam is represented by a finite sized array of discrete points, the phase of the beam cannot be accurately represented if the phase of the angular spectrum propagator changes too rapidly between points.

When using the angular spectrum propagator, the phase of the electric field is measured relative to a plane. Positive phase indicates the wavefront is advanced along the local +z axis relative to the plane, regardless of the direction of propagation.

The angular spectrum propagator is useful when the Fresnel number is large. This includes the important case of propagating a beam a short distance. However, the angular spectrum propagator also works well for propagating a large distance when the divergence of the beam (and thus ρ) is small. A good rule of thumb to use is that if the beam does not change size significantly, the angular spectrum propagator may be used. To propagate beams with small Fresnel numbers, where the beam will change size significantly, requires a separate theoretical and numerical method.

Fresnel diffraction

For small Fresnel numbers, the appropriate theory is Fresnel diffraction. For a complete discussion and derivation, see the Goodman reference given in the introduction. The key assumptions in Fresnel theory require
that the field being computed is not too close to the initial field, namely, if z2 – z1 = Δz , then Δz is large compared to the region over which the field at z_2 is to be determined. Another way of saying this is that the beam cannot diverge too quickly; very fast F/# beams cannot be accurately modeled with Fresnel diffraction theory (for more information on these approximations search the help files for "Algorithm assumptions").

In the Fresnel region the electric field distribution is given by

$$E(x_2, y_2, z_2) = \left[\frac{e^{ikz}}{i\lambda\Delta z}\right] q(r_2, \Delta z) \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E(x_1, y_1, z_1)q(r_1, \Delta z)e^{-\frac{i2\pi}{\lambda\Delta z}(x_1x_2 + y_1y_2)} dxdy ,$$

where $\quad q(r, \Delta z) = e^{(i\pi r^2)/(\lambda\Delta z)} .$

Each of the terms in the above expression has a useful physical interpretation. The leading term indicates that as the beam propagates, the phase changes along the z axis, just like the plane wave described earlier. The amplitude also decreases linearly with distance, or the intensity (E*E) falls quadratically. The expression for q(r, Δz) , called the quadratic phase factor, indicates that the phase is referenced to a sphere of radius Δz (strictly speaking it is a parabola, but we have already assumed in the Fresnel development that r2 « Δz). This is a very useful property; all that is required in the representation of the electric field is the phase difference relative to the reference sphere. This significantly reduces the number of sample points needed to accurately define the phase of the beam.

When using the Fresnel propagator, the phase of the electric field is measured relative to a reference sphere with a radius equal to that of the distance from the beam waist. This is not the same radius as the phase radius of curvature of the Gaussian beam. Positive phase indicates the wavefront is advanced along the local +z axis relative to the reference sphere, regardless of the direction of propagation.

Another important property of q(r, Δz) is that as Δz gets larger, q(r, Δz), varies more slowly in phase. This is the opposite of the T(Δz) operator, which varies rapidly in phase as Δz gets larger. Accordingly, Fresnel diffraction is useful when the Fresnel number is small.

Selecting the correct propagator

Zemax will automatically choose the angular spectrum propagator when the Fresnel number is large, and the Fresnel propagator when the Fresnel number is small. However, there are times when the angular spectrum propagator is a better choice than the Fresnel, and Zemax supports a surface specific option to choose the angular spectrum propagator rather than the default choice.

Fraunhofer diffraction

Consider the Fresnel diffraction expression. If Δz is very large, then q(r, Δz) may be neglected. This yields the Fraunhofer diffraction expression, which is

$$E(x_2, y_2, z_2) = \left[\frac{e^{ikz}}{i\lambda\Delta z}\right] \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} E(x_1, y_1, z_1)e^{-\frac{i2\pi}{\lambda\Delta z}(x_1 x_2 + y_1 y_2)}dxdy,$$

or

$$E(x_2, y_2, z_2) = FF(E(x_1, y_1, z_1)),$$

where phase and amplitude factors are omitted. The far field distribution is just a scaled version of the Fourier transform of the near field distribution. Fraunhofer diffraction is only valid if the Fresnel number is nearly zero.

The ray based diffraction features in Zemax, such as the diffraction MTF and PSF assume Fraunhofer diffraction. This is why Zemax cannot compute the ray based diffraction MTF or PSF if the beam is too much out of focus. The Fraunhofer assumption is never used by the physical optics propagation algorithm in Zemax, it is presented here for completeness.

The pilot beam

Consider a Gaussian beam with waist ω_0 . The Rayleigh range is given by

$$z_r = \frac{\pi\omega_0^2}{\lambda}.$$

The phase radius of curvature of the beam is a function of the distance from the beam waist, z:

$$R(z) = z + \frac{z_r^2}{z}.$$

Note that radius is infinite at z = 0, reaches a minimum of $2z_r$ at z = z_r , and asymptotically approaches infinity as z approaches infinity. The phase of the Gaussian beam along the axis is defined by the Gouy shift, given by

$$\theta(z) = \tan^{-1}\left(\frac{z}{z_r}\right).$$

For example, the axial phase is $\pi/4$ at a distance of plus one Rayleigh range. The beam size is also a function of the distance from the waist:

$$\omega(z) = \omega_0 \left[1 + \left(\frac{z}{z_r} \right)^2 \right]^{1/2}.$$

Note for large distances the beam size expands linearly. The divergence angle of the beam is given by

$$\theta = \tan^{-1} \frac{\omega_0}{z_r}.$$

Now consider the problem of numerically representing this beam by a discrete sampling of points. If a constant spacing between points is used, the beam will expand beyond the edges of the array if the propagation proceeds too far from the waist. Therefore, far from the waist, a linearly expanding coordinate system where the point spacing is proportional to z is best. However, near the waist, the beam size does not decrease to zero, but remains reasonably constant. In this domain, a constant sampling is most convenient. The compromise sampling system is to use a constant spacing near the waist, and a linearly scaled spacing far from the waist.

The diffraction theory developed earlier did not assume any particular shape or form to the electric field being propagated. The algorithms are for the most part independent of the field distribution. However, the problem of sampling remains. It is also impractical to compute (or even define) Fresnel numbers for arbitrary, aberrated beams with irregular or non-existent apertures.

For this reason, a pilot beam is used to assist the physical optics propagation algorithm in determining which propagation algorithm to select. The pilot beam is an ideal Gaussian beam, with a waist, beam size, phase radius, and relative z position. The initial parameters may be generated by fitting the Gaussian beam equations to the initial distribution. The pilot beam is then propagated from surface to surface. At each surface, new beam parameters, such as the new waist, phase radius, or position are computed. The properties of the pilot beam are then used to determine if the actual distribution is inside or outside the Rayleigh range, and what propagation algorithms are appropriate.

After passing through an aperture that significantly truncates the beam, such as a pinhole aperture, it may be required to recompute the pilot beam parameters, as described later in this help file.

Sign conventions for phase data

It is very important to understand the phase sign conventions when constructing beams using DLL files, beam files, or when performing fiber coupling calculations where phase matching is critical.

As shown in the previous sections, the angular spectrum propagator works best when the beam is nearly collimated, while the Fresnel theory works best when the beam is diverging. When using the angular spectrum propagator, the phase of the elec- tric field is referenced to a plane.

Once the beam propagates past the Rayleigh range, the Fresnel propagator is used, and the phase of the electric field is referenced to a sphere whose radius is the distance from the beam waist to the current position of the pilot beam.

The sign of the phase is positive if the wavefront is to the "right" of the reference surface, with "right" being towards the positive local Z axis direction. For a beam propagated to the +z side just inside the Rayleigh range, the phase slope is negative be- cause the wavefront is left of the reference plane. Just outside of the Rayleigh range, the phase slope is positive because the wavefront is now to the right of the reference sphere.

Therefore, the slope of the phase of the electric field will "flip" from negative to positive when cross- ing from inside to outside the Rayleigh range on the +z side.
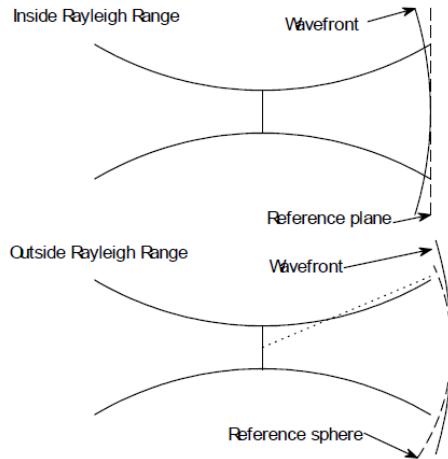
For example, consider a Gaussian beam. At the waist, the phase is zero everywhere on the reference plane. If the beam is propagated to just inside the Rayleigh range on the positive side of the waist, the phase at the center of the beam will change to $\pi/4$ radians, this is the Gouy shift given by $\varphi = \tan^{-1}(z/z_r)$. The phase radius of curvature of the beam will become twice the Rayleigh range distance. The wavefront phase will be increasingly negative relative to the center of the wavefront as the radial aperture is increased, because the curved wavefront phase is measured relative to a plane.

If the beam now propagates a small distance so the pilot beam is now just on the far side of the Rayleigh range, the phase will remain $\pi/4$ radians at the center but will now be referenced to a sphere whose radius is the distance to the waist, which is the Rayleigh range for this case. Because the radius of the reference sphere is smaller than the curvature of the wavefront, the slope of the phase of the beam will now "flip" to be increasingly positive with radial aperture.

Finally, as the beam propagates further past the Rayleigh range, the phase relative to the reference sphere will tend toward a constant value of $\pi/2$ radians (the limiting value of the Gouy shift).

The two representations of the beam are equivalent, however care must be taken to account for the proper phase reference surface when constructing or comparing different beams.

Propagating in and out of the Rayleigh range

The naming conventions used in this section are from the Lawrence reference cited earlier in this help file.

It has already been shown that short propagations are well modeled using the PTP operator. This operator has the property that the sample spacing remains constant.

It is most convenient to use the Fresnel propagator when the beam is already at the waist of the pilot beam, and the field is desired far from the waist relative to a reference sphere. For this reason, the Fresnel propagator is redefined as the waist to sphere (WTS) operator:

$$E(x_2, y_2, z_2) = WTS[E(x_1, y_1, 0), z_2] \quad , \text{ where}$$

$$WTS(E, \Delta z) \equiv \left[\frac{1}{i\lambda z_2}\right] FF^s[q(r_1, z_2)E], \ s = \frac{z_2}{|z_2|},$$

$$\Delta x_2 = \frac{\lambda|z_2|}{n_x \Delta x_1}, \text{ and } \Delta y_2 = \frac{\lambda|z_2|}{n_y \Delta y_1},$$

where nx and ny are the number of points in the x and y directions of the array. The last two expressions yield the new linearly scaled sample spacings after application of the operator. Reversing the order of operations results in the sphere to waist (STW) operator:

$$STW(E, \Delta z) \equiv q(r_1, z_2)FF^s[i\lambda z_2 E],$$

with a similar change in the sample spacing. There are four possible general propagation cases to consider:

    II: propagation from inside to inside the Rayleigh range.
    IO: propagation from inside to outside the Rayleigh range.

OI: propagation from outside to inside the Rayleigh range.
OO: propagation from outside to outside the Rayleigh range.

All these cases can be handled with the appropriate combination of the PTP, WTS, and STW operators:

II(z1, z2) = PTP(z2 - z1)
IO(z1, z2) = WTS(z2 - z0)PTP(z0 - z1)
OI(z1, z2) = PTP(z2 - z0)STW(z0 - z1)
OO(z1, z2) = WTS(z2 - z0)STW(z0 - z1)

where z0 is the pilot beam waist position, z1 is the starting beam position, and z2 is the end beam position.

Separation of X and Y propagation

For rotationally symmetric optics, a single rotationally symmetric pilot beam and propagation method is appropriate. When the beam is astigmatic or anamorphic, and/or the optical components are cylindrical or toroidal, then a more efficient model for the beam is to separate the X and Y propagation. Separation of X and Y direction propagation is accomplished by choosing the appropriate option in the General Tab of the Physical Optics Propagation analysis window settings.

If the propagation in X and Y is separated, then there will be two pilot beams; one each for the X and Y direction. The phase references will also be distinct in the X and Y directions. In place of a spherical reference surface, a "bispherical" surface is used instead. The sag of a bispherical surface is given by:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - c_x^2 x^2 - c_y^2 y^2}}, \quad \text{where}$$

$$c_x = \frac{1}{R_x}, \, c_y = \frac{1}{R_y}.$$

The pilot beam, including beam phase radius, waist, position, Rayleigh Range, divergence, are all distinct in the X and Y directions. There is a small amount of additional computational overhead in propagating the X and Y portions of the beam independently, and so this option should only be used when required.

Comments about point spacing and sampling

Although the total number of array points n_x and n_y, remain constant, the array size and point spacings Δx and Δy will change as the beam propagates. If the array width is very large at the beam waist relative to the waist size, then there are relatively few points across the beam waist. This will result in a smaller array size far from the waist, with a relatively large number of points across the beam size. Conversely, if the array size is small at the waist, the array size will grow large compared to the beam far from the waist, leaving few sample points to represent the beam. This inverse relationship is a necessary but frequently inconvenient product of the Fourier transform theory used to model the diffraction. The exact equations describing the change in point spacing are given in the previous section.

There is clearly a tradeoff between good sampling of the beam near the waist and good sampling far from the waist. It can be shown that to achieve approximately uniform sampling relative to the beam size at both the waist and far from the beam waist the array size at the beam waist in the X and Y directions should be

$$X = \omega_{0x}\sqrt{\pi n_x}, \, Y = \omega_{0y}\sqrt{\pi n_y}.$$

The physical optics analysis feature settings include an "Auto" button which will use this formula to set a suggested initial array width. For more information, search the Analyze Tab section of the help files for "Physical Optics Propagation"

Propagation through arbitrary optical surfaces

The methods described thus far are good for propagating through homogeneous space. However, the prime interest is in propagating the beam through optical components such as lenses and apertures.

It is not practical to directly perform a diffraction propagation through an arbitrary surface shape. The difficulty lies in the representation of the complex amplitude at discrete points in a plane or spherical phase referenced array. When the beam is incident on a curved surface, different parts of the beam intercept the surface at different points along the local z axis.

To avoid this problem, the properties of the pilot beam are used to generate a set of rays that represent the wavefront incident upon the surface. This set of rays, called the probing rays, is then traced through the optical surface using conventional geometric optics. The path length of each ray, and the positional and angular aberrations generated, are then used to reconstruct the pilot beam and complex amplitude after the surface. The probing ray set is used to generate the transfer through the surface.

The probing ray set can be used to determine:

The effective power of the surface with respect to the pilot beam. The direction of the beam after leaving the surface.

The polarization phase and amplitude transmission of a surface.

The vignetting of the surface with any supported surface apertures (or ray errors). Note that surfaces without hard apertures defined will pass the entire beam without vignetting, independent of how the system aperture is set.

The phase aberrations.

The anamorphic and non-linear stretching and compression (distortion) of the beam.

The probing ray method can be applied to propagation through a single surface, or multiple surfaces at once. This is a very desirable property, because geometrical optics may be used to propagate through whole optical components that would be difficult to model with physical optics propagation. These include highly tilted surfaces and gradient index lenses, to name a few. Propagating through multiple surfaces at once using rays also speeds up the analysis. Some special surfaces, such as the ABCD matrix surface, and some types of Fresnel surfaces, do not allow POP analysis at all because there is no way to compute the effective phase of these surface types. A warning is issued by Zemax if the POP analysis cannot proceed due to the presence of these special surface types.

Propagating through non-sequential surfaces

Non-sequential component (NSC) surfaces pose a difficult challenge for physical optics propagation. A NSC surface may represent a large number of discrete components. The incoming beam may not hit all or any of these components, or perhaps a portion of the beam will hit one component and split off from the rest of the beam, or any number of other possibilities. The Fresnel propagation method used by POP is not suitable for computing the effects of propagation through arbitrary NSC surfaces.

There are a limited number of cases where POP can be used to propagate a beam through a NSC surface with reasonable confidence in the results. These include:

-NSC surfaces with a single lens or mirror.

-Prisms with an effectively sequential path and apertures large compared to the beam.

Even in these cases, POP can only use ray tracing to model the effective phase the NSC surface imparts to the beam. Diffraction propagation is not possible. For this reason, only relatively slow beams propagating through thin NSC surfaces should be used. Always check the POP results carefully before having confidence in results of propagation through NSC surfaces.

Accounting for polarization

If polarization is used, Zemax will use polarization ray tracing to determine the properties of the probing ray set and corresponding transfer function. Polarization ray tracing permits the modeling of the effects of optical coatings on the phase and amplitude of the transmitted or reflected beam.

Zemax assumes the Ex and Ey portions of the field account for all the energy in the beam at any given array point. However, the Ez component is required for polarization ray tracing. Zemax recreates the Ez as needed by applying the condition that E must be perpendicular to the propagation vector k. The vector k is computed from the pilot beam

properties and the phase errors present on the beam. The Ex and Ey fields are renormalized to account for the correct beam intensity. After polarization propagation, the Ex and Ey components are renormal- ized to again hold the energy present in the Ez component, which is then discarded.

Memory requirements

Geometric optics ray tracing generally requires very little RAM, because each ray may be traced from object to image, independently of all other rays. Physical optics is far more demanding, because the entire beam must be stored at once. Zemax uses 8 byte "double precision" floating point numbers to store the beam data. There are actually 16 bytes per array position because the electric field amplitude is a complex number. Here are some guidelines for how much RAM is required for various parts of the propagation algorithm:

Task	RAM Requirement
Storing an unpolarized beam	16 bytes per grid point
Storing a polarized beam	32 bytes per grid point
Computing the surface transfer probing ray set	56 bytes per grid point
	Some special surface types such as paraxial and birefringent surfaces require additional data and may require up to 90 bytes per grid point

For example, propagating a 2048 x 2048 unpolarized beam through normal refractive surfaces will require about 288 Mb. Zemax currently supports a minimum array size of 32 x 32 and a maximum array size of 8192 x 8192 for 32-bit versions of Zemax and 16,384 x 16,384 for 64-bit versions of Zemax. Zemax requires the sampling to be an integral power of 2; for example, 32, 64, 128, 256, etc.

If not enough RAM is available, the local hard disk drive may be used for temporary storage. If the option "Use Disk Storage To Save Memory" is checked, Zemax will store the surface transfer array (at 56 bytes per grid point) in a temporary file. Although this saves a substantial amount of RAM, the analysis is much slower. Typically, using the disk storage option will slow the analysis by a factor of 10 to 20. If the computer does not have sufficient RAM, it may automatically "page" to the hard disk. This will allow the computation to proceed, but at a rate hundreds or even thousands of times slower than if sufficient RAM were available.

32-bit versions of Windows do not allow Zemax to allocate more than 3.0 Gb of RAM, and in many cases, the actual limit is much lower depending upon how much actual RAM the computer has. This limit is imposed by Windows, and not Zemax. Both Zemax and Windows are available as 64-bit versions, and these versions allow use of memory well beyond 3.0 Gb. For complex POP calculations, use of 64-bit Windows and Zemax is strongly recommended.

Defining the initial beam

The menu option Analysis, Physical Optics, Physical Optics Propagation invokes the Physical Optics Propa- gation feature. The settings box for this feature allows user selection of many options, including those needed to define the initial beam, sampling, surface ranges, and field positions. Search the Analyze Tab section of the help files for "Physical Optics Propagation" for a discussion of the available options and settings.

The X- and Y-Sampling determine the number of points used to sample the beam. Larger values are generally more accurate at the expense of longer computation times and larger RAM requirements.

The X- and Y-Width are measured in lens units. The larger the width, the more empty space around the portion of the beam with non-zero intensity there will be. This space is called the guard band, and it is vital that sufficient empty space exist around the beam. The empty space allows room for the beam to expand as aberrations are introduced to the beam. If portions of the beam get too close to the edge of the array, they will "alias" and reflect back into the beam, decreasing the accuracy of the computed results.

The initial beam may be any of these types: Gaussian Waist, Gaussian Angle, Gaussian Size+Angle, Astigmatic Gaussian, Top Hat, File, DLL, or Multimode. See the following sections for a detailed discussion of each beam type. The beam may then be aligned along the chief ray for any defined field position, in the optical space preceding any surface. The starting location for the beam may also be offset from the starting surface by the "Surf To Beam" setting (search the Analyze Tab section of the help files "Physical Optics Propagation"). Propagation then proceeds to the

ending surface. Definitions of beam waist, Rayleigh range, and divergence angle may be found in "Propagating the embedded beam".

Gaussian Waist

The Gaussian Waist beam is an optionally truncated and decentered Hermite-Gaussian of arbitrary order defined by:

$$E(x, y) = E_0 T(x, y) G_l \left[\frac{\sqrt{2}(x - dx)}{\omega_{0x}} \right] G_m \left[\frac{\sqrt{2}(y - dy)}{\omega_{0y}} \right], \text{where}$$

$$G_i(u) = H_i(u) \exp\left(\frac{-u^2}{2} \right),$$

And H_i (u) is the Hermite polynomial function of order i. Note the order may be specified in the x and y directions independently. The x order is defined by the integer l, and the y order is defined by the integer m. If both l and m are zero, the simple "Gaussian" TEM(0,0) beam is generated. Higher order modes may be generated by modifying the order values; for example, to generate TEM(1,2) set l = 1 and m = 2. For a discussion of Hermite- Gaussian beams see Saleh, B. E. A., and Teich, M. C., Fundamentals of Photonics, John Wiley & Sons, New York (1991). If the order number is set higher than 30, the order zero will be used to prevent excessively long computation time.

The dx and dy values are used to decenter the beam. The transmittance function T (x, y) is used to (optionally) truncate the beam to a finite aperture. The transmittance function is defined as

$$T(x, y) = 1, \text{ if } \left(\frac{x - dx}{A_x} \right)^2 + \left(\frac{y - dy}{A_y} \right)^2 \leq 1,$$

and

$$T(x, y) = 0 \quad \text{otherwise.}$$

Ax and Ay are the truncating aperture values. If Ax or Ay are zero, no truncating aperture is used. A smoothing function is used near the edge of the truncating aperture to reduce pixel related errors. The smoothing function weights the pixel amplitude by the area of the pixel inside the truncating aperture. The truncating aperture is useful for modeling receiver fiber modes, where the truncating aperture is typically 15% greater than the core size.

The value for E_0 is chosen to yield the peak irradiance in power per unit area or the total beam power as defined in the settings box.

The beam is defined at the waist and will generally diverge to a larger beam size as it propagates away from the waist. See Gaussian Angle and Gaussian Size+Angle below.

Gaussian Angle

The Gaussian Angle beam is an optionally decentered TEM(0, 0) mode whose waist size is determined by the far-field divergence half-angle in degrees, measured in air. Zemax uses the specified divergence angle to compute the beam waist. The beam waist is given by:

$$\omega_0 = \frac{\lambda}{\pi \tan \theta}.$$

Note that the angle is the half-angle of divergence in degrees, and the wavelength is not scaled by the index of the starting medium. If the x and y angle values are different, then an elliptical beam will be generated. The beam is defined at the waist.

Gaussian Size+Angle

The Gaussian Size+Angle beam is an optionally decentered TEM(0, 0) mode defined by the beam size (not waist) and the far-field divergence half-angle in degrees, measured in air. Zemax uses the specified beam size and divergence angle to compute the beam waist, position, and phase. The beam waist is given by:

$$\omega_0 = \frac{\lambda}{\pi \tan\theta}.$$

Note that the angle is the half-angle of divergence in degrees, and the wavelength is not scaled by the index of the starting medium. If the resulting beam waist is larger than the specified beam size (a physical impossibility), the waist value will be used for the beam size. The z position of the beam relative to the waist is then computed using:

$$z = z_r \sqrt{\left(\frac{\omega}{\omega_0}\right)^2 - 1},$$

where z_r is the Rayleigh range. If the x and y direction values for the various data are different, then an elliptical beam with a possibly toroidal phase will be generated.

Astigmatic Gaussian

The Astigmatic Gaussian beam is an optionally decentered TEM(0,0) mode defined by the beam waist values in X and Y and the beam waist positions in X and Y. The beam waist positions are the distances from the current beam position (defined at the Start Surface and accounting for any non-zero Surface To Beam value) to the beam waists in X and Y (the value is negative if the waist is to the left of the current beam position).

Note that the Astigmatic Gaussian beam is functionally identical to the Gaussian Size+Angle beam described above. The Astigmatic Gaussian beam option is simply provided as an alternative to the Gaussian Size+Angle beam option for applications in which the waist values and waist positions of the beam are known rather than the beam size and far-field divergence angle.

Top Hat

The top hat beam is an optionally decentered uniform amplitude beam defined as

$$E(x, y) = E_0 \text{ if } \left(\frac{x - dx}{w_x}\right)^2 + \left(\frac{y - dy}{w_y}\right)^2 \leq 1, \text{ and}$$

$$E(x, y) = 0 \text{ if } \left(\frac{x - dx}{w_x}\right)^2 + \left(\frac{y - dy}{w_y}\right)^2 > 1,$$

where E_0 is chosen to yield the peak irradiance in power per unit area or the total beam power as defined in the settings box.

File

A beam may be defined in a user defined table properly formatted into a file. The table of values must be placed in either a binary or text format file and read from disk. The file must end in the extension ZBF (for Zemax Beam File). The binary format is identical to the format written out by Zemax if the "Save Output Beam To:" option is selected. The Ex and Ey values are defined such that the Ex*Ex + Ey*Ey is in units of watts. If the units flag indicates the beam units are different from the current lens units, the beam is automatically scaled to the current lens units when read into Zemax.

All ZBF Beam Files must be placed in the \<pop\> folder (for details, search the help files for "Folders").

Beams in Zemax are always centered on the chief ray for the selected field and wavelength. Therefore, the data in the beam file should be positioned relative to the chief ray that will be used to align the beam. The center point in the beam file is at the coordinate (nx/2+1, ny/2+1). Zemax requires the values of nx and ny to be an integral power of 2; for example, 32, 64, 128, 256, etc. The minimum sampling is 32 and the current maximum sampling is 8192 for 32-bit versions of Zemax and 16,384 for 64-bit versions of Zemax. Fiber coupling data is ignored when reading beams, and will be zero if fiber coupling is not computed on output. Note the total fiber coupling is the product of the receiver and system efficiency. The first data point is at the -x, -y corner, and the data proceeds across the x rows first. The Rayleigh distance is ignored on input and is automatically recomputed by Zemax. The wavelength value stored in the ZBF file is scaled by the index of the media the beam is currently in.

Zemax Beam File (ZBF) binary format

The ZBF binary file format is defined as follows. All integers are 4 bytes, all doubles are 8 bytes.

```
1 integer: The format version number, currently 1.
1 integer: The number of x samples (nx).
1 integer: The number of y samples (ny).
1 integer: The "is polarized" flag; 0 for unpolarized, 1 for polarized.
1 integer: Units, 0 for mm, 1 for cm, 2 for in, 3 for meters.
4 integers: Currently unused, may be any value.
1 double: The x direction spacing between points.
1 double: The y direction spacing between points.
1 double: The z position relative to the pilot beam waist, x direction.
1 double: The Rayleigh distance for the pilot beam, x direction.
1 double: The waist in lens units of the pilot beam, x direction.
1 double: The z position relative to the pilot beam waist, y direction.
1 double: The Rayleigh distance for the pilot beam, y direction.
1 double: The waist in lens units of the pilot beam, y direction.
1 double: The wavelength in lens units of the beam in the current medium.
1 double: The index of refraction in the current medium.
1 double: The receiver efficiency. Zero if fiber coupling is not computed.
1 double: The system efficiency. Zero if fiber coupling is not computed.
8 doubles: Currently unused, may be any value.
2*nx*ny double: Ex values.
If polarized, 2*nx*ny Ey values follow the Ex values.
```

Zemax Beam File (ZBF) text format

The ZBF text file format is defined as follows. The first line must be the single character "A", followed by the other data values specified.

```
A: indicates a text file.
```

version: The format version number, currently 1.

nx: The number of x samples.

ny: The number of y samples.

ispol: The "is polarized" flag; 0 for unpolarized, 1 for polarized.

units: 0 for mm, 1 for cm, 2 for in, 3 for meters.

unused 1: Currently unused, may be any value.

unused 2: Currently unused, may be any value.

unused 3: Currently unused, may be any value.

unused 4: Currently unused, may be any value.

dx: The x direction spacing between points.

dy: The y direction spacing between points.

zx: The z position relative to the pilot beam waist, x direction.

Rx: The Rayleigh distance for the pilot beam, x direction.

wx: The waist in lens units of the pilot beam, x direction.

zy: The z position relative to the pilot beam waist, y direction.

Ry: The Rayleigh distance for the pilot beam, y direction.

wy: The waist in lens units of the pilot beam, y direction.

lambda: The wavelength in lens units of the beam in the current medium.

index: The index of refraction in the current medium.

re: The receiver efficiency. Zero if fiber coupling is not computed.

se: The system efficiency. Zero if fiber coupling is not computed.

unused 1: Currently unused, may be any value.

unused 2: Currently unused, may be any value.

unused 3: Currently unused, may be any value.

unused 4: Currently unused, may be any value.

unused 5: Currently unused, may be any value.

unused 6: Currently unused, may be any value.

unused 7: Currently unused, may be any value.

unused 8: Currently unused, may be any value.

Ex real value for point 1

Ex imaginary value for point 1

Ex real value for point 2

Ex imaginary value for point 2

etc... for 2*nx*ny Ex values.

If polarized, followed by 2*nx*ny Ey values.

DLL

To define a beam using an external program, the algorithm which computes the initial complex electric field from user defined parameter data must be written and compiled into a Windows Dynamic Link Library, or DLL. Sample DLLs are provided with Zemax with source code. Studying an existing DLL is the best way to see how the DLL is used to define the beam amplitude and phase. New DLLs may be easily created with a suitable compiler.

Beam DLL parameters

Each DLL may use between zero and 8 user defined data values as parameters in the computation of the beam properties. These values are defined by the DLL and are only used by the DLL. The values may be entered or edited directly on the settings box when the DLL is selected as the beam type.

Creating a new Beam DLL

The DLL must include two functions: UserBeamDefinition UserParamNames

When initializing a beam using a DLL, Zemax passes to the UserBeamDefinition function the source parameters, the wavelength, and other data. The UserBeamDefinition function then is required to compute the electric field values. These values are returned to Zemax and are used to populate the beam array. The DLL also returns values for the pilot beam, including the waist size, position, and Rayleigh range in the x and y directions. If all of these pilot beam values are zero, Zemax will automatically fit the pilot beam parameters to the electric field data.

The function UserParamNames is used to define the names of all used parameters. These names appear on the settings dialog box.

The best way to learn the use of Beam DLLs is to copy and study an existing DLL. The sample DLLs provided with Zemax include extensive documentation and comments on the data format; see any of the sample source code files for examples.

All Beam DLLs must be placed in the <program>\DLL\PhysicalOptics folder.

Multimode

A multimode beam consists of a sum of any number of other beams. The sum may be coherent or incoherent, and several options exist for scaling, randomizing, and altering the phase of the individual components of the beam. The multimode beam sum is defined in a user defined text file. The file must end in the extension ZMM (for Zemax Multi Mode).

All ZMM Files must be placed in the <pop> folder (for details, search the help files for "Folders").

The ZMM file uses a simple command syntax that defines the multimode beam. The "master" beam is initially set to zero amplitude over the entire beam. The individual component beams are generated in the sequence defined in the ZMM file. Each beam as defined in the ZMM file is then summed either coherently or incoherently with the master beam. The sum mode, either coherent or incoherent, is controlled by the COHERENT and INCOHERENT commands. The default sum mode is COHERENT. All of the available ZMM commands are described below.

!, Comment

Any data following the ! symbol is ignored. Comment lines may be used by starting the line with a ! symbol. Blank lines are also allowed and are ignored.

COHERENT

Sets the beam summing mode to coherent. The syntax is:

```
COHERENT
```

The coherent sum of any two beams is computed by summing the point-by-point real and imaginary parts of the two beams. The phase of the resulting beam thus depends upon the vector sum of the components. All subsequent beam sums will be coherent unless an INCOHERENT command is used. See also INCOHERENT. The default sum mode is COHERENT.

DLL

The DLL command defines a beam based upon a user-defined DLL, which is then summed to the master beam. The syntax is:

```
DLL weight DLL_NAME param1 param2 param3 ...
```

Note that the data values are separated by spaces. The weight value defines the dimensionless relative total power of this beam as compared to other beams in the same ZMM file. The weight value may be negative, for details search the help files for "Using random values". The DLL_NAME is the name of the DLL file (with no extension). The DLL file must be placed in the correct folder, see "DLL" for details. The other values are exactly as defined for the DLL parameters. No scaling of the parameters is supported, since the dimensionality and purpose of the DLL parameters cannot be known in advance.

GW

The GW command defines a Gaussian Waist beam which is then summed to the master beam. The syntax is:

```
GW weight waistx waisty decenterx decentery aperturex aperturey orderx ordery
```

Note that the data values are separated by spaces. The weight value defines the dimensionless relative total power of this beam as compared to other beams in the same ZMM file. The weight value may be negative, for details search the help files for "Using random values". The other values are exactly as defined for the Gaussian Waist beam, search the help files for "Gaussian Waist" for a description. The transverse data values, such as the waist and aperture sizes, may be scaled from the POP Beam Definition tab; search the help files for "Using the scale factor" for details.

INCOHERENT

Sets the beam summing mode to incoherent. The syntax is:

```
INCOHERENT
```

The incoherent sum of any two beams is computed by summing the point-by-point intensity of the two beams, taking the square root to determine the amplitude, and setting the real part of the combined beam to this amplitude, and the imaginary part to zero. All phase information, including that of aberrations in the beam, is lost when performing incoherent sums. Therefore care should be taken when using this command. All subsequent beam sums will be incoherent unless a COHERENT command is used. See also COHERENT. The default sum mode is COHERENT.

PHASE

The PHASE command multiples any subsequently defined beams by a complex factor determined by the defined phase angle. The syntax is:

```
PHASE angle
```

or

```
PHASE RANDOM
```

The angle is in units of degrees and may be any value. If the angle is set to the keyword RANDOM, the phase is randomly selected to be between -180 and 180 degrees. Note the random angle generated depends upon the random seed value used, search the help files for "Using random values" for details. All subsequently defined beams are multiplied by the resulting complex factor before being added to the master beam.

TH

The TH command defines a Top Hat beam. The syntax is:

```
TH weight waistx waisty decenterx decentery
```

Note that the data values are separated by spaces. The weight value defines the dimensionless relative total power of this beam as compared to other beams in the same ZMM file. The weight value may be negative, for details, search the help files for "Using random values". The other values are exactly as defined for the Top Hat beam, search the help files for

"Top Hat" for a description. The transverse data values, such as the waist and aperture sizes, may be scaled from the POP Beam Definition tab; search the help files for "Using the scale factor" for details.

Using random values

The weight value used by some of the beam commands may be negative, which means a random weight will be assigned to the beam. The random weight will be between 0.0 and the absolute value of the weight. For example, if the weight is -2.0, then a random value will be assigned to the weight between 0.0 and 2.0.

The PHASE RANDOM command uses the random number generator to compute a random phase.

The random values used will always be the same given the same initial "seed" value. The seed value for the random number generator is a user definable value and is Parameter 1 for the multimode beam type. Use a different seed value to generate different, random, but reproducible beams.

Using the scale factor

It is convenient to be able to scale a beam without changing the fundamental data as defined in the ZMM file. Parameter 2 of the Multimode beam type as defined on the POP Settings Beam Definition tab is called "scale". This value, if not zero, will be used to scale all transverse data for all beams in the file.

Surface specific settings

Each Zemax surface supports these settings relevant to Physical Optics Propagation:

Use Rays To Propagate To Next Surface

Do Not Rescale Beam Size Using Ray Data

Use Angular Spectrum Propagator

Draw "lens file name-surface #" on shaded model

Re-Compute Pilot Beam Parameters

Use X-axis Reference

Resample After Refraction

Auto Resample

Output Pilot Radius

For more information on these settings, see the Physical Optics section of the Surface Properties in the Lens Data Editor.

Considerations when using rays to propagate

When using "Use Rays To Propagate To Next Surface", the POP algorithm must first convert the wavefront to a set of rays that represents the beam. The rays are then traced through all consecutive surfaces that have this option checked on. When the rays reach the first surface with this option checked off, the rays must then be converted back to a beam. If the rays form a focus on this surface, the conversion back to a beam cannot be done accurately, because the rays do not adequately represent the diffracted beam near focus.

The solution is to use an additional (usually dummy) surface to propagate the rays some distance away from focus (at least one Rayleigh range), and then end the "Use Rays" section of the propagation at the dummy surface. The rays can then be accurately converted to a wavefront and the propagation can proceed, even if a virtual propagation back to the focus is necessary.

Computing Fiber Coupling

The physical optics propagation algorithm may be used to compute fiber coupling efficiency. A ray based method is also supported, for details search the help files for "Fiber Coupling Efficiency".

The fiber coupling receiver efficiency is defined as a normalized overlap integral between the fiber and beam complex amplitude:

$$T = \frac{\left| \iint F_r(x, y) W'(x, y) dx dy \right|^2}{\iint F_r(x, y) F_r'(x, y) dx dy \iint W(x, y) W'(x, y) dx dy},$$

Where $F_r (x, y)$ is the function describing the receiving fiber complex amplitude, $W(x, y)$ is the function describing the complex amplitude of the beam coupling into the fiber, and the ' symbol represents complex conjugate. Note that these functions are all complex valued, so this is a coherent overlap integral. Maximum receiver efficiency ($T = 1.0$) is achieved when the mode of the beam perfectly matches the mode of the fiber in both amplitude and phase at all points. Any deviation in mode shape, or phase, will reduce the value of T to less than 1.0. Optical aberrations typically introduce phase deviations which reduce receiver efficiency.

Additional system losses may be caused by apertures which vignette the beam, losses due to reflection from air-glass boundaries, or bulk absorption. These loss factors reduce the system efficiency. The total coupling efficiency into the fiber is the system efficiency multiplied by the receiver efficiency.

Where the integral is computed

The fiber coupling receiver efficiency integral defined above is valid and may be computed anywhere in the optical system; including at the location of the fiber, or at some location near or far from the fiber. The **critical requirement** is that the fiber mode function describes the mode of the fiber at the point the integral is computed, which is always the "end" surface (search the Analyze Tab section of the help files for "Physical Optics" for a list of analysis options, including specification of the end surface and the fiber mode). If the end surface represents the beam at the location of the receiving fiber, then the fiber mode parameters must correspond to the mode at the receiving fiber. If the end surface is 10 mm away from the fiber, then the fiber mode parameters must be selected to define the mode 10 mm away from the fiber.

Defining the fiber mode

The fiber mode may be a Gaussian or Top Hat function, or may be defined by a DLL or a data file. This allows very general and arbitrary fiber modes to be described, including multi-mode, aberrated, or arbitrary amplitude and phase fibers. The fiber mode may be defined using all the same options allowed for defining initial beams; for details, search the help files for "Defining the initial beam".

If the beam is not polarized, then the fiber mode is not polarized and only the X-direction of the E field of the fiber mode is used in the overlap integral. If the beam is polarized, then the fiber mode may be either polarized or unpolarized. If the setting "Ignore Polarization" on the Fiber Data Tab (for details, search the Lasers and Fibers section of Analyze Tab help file for "Fiber Data Tab") is checked, then the fiber mode is unpolarized, and the X-direction E field is used to compute

the coupling for both the X- and Y-direction fields in the polarized beam. If "Ignore Polarization" is unchecked, then the coupling for the X- and Y-direction is done independently and then combined to yield the overall coupling.

Decenters and tilts

It is usually valuable to determine fiber coupling for decentered and tilted fibers.

Decenter X/Y values are in lens units and may be defined on the fiber data tab of the Physical Optics Propagation settings box ("Physical Optics Propagation"). These values decenter the fiber mode. The decenter values shift the center of the mode within the array defining the fiber mode at the end surface. For this reason, the decenter values should be small enough to keep the majority of the mode energy inside the array width/height at the end surface.

The tilt values are in degrees. Tilt of the fiber is modeled as adding a linear phase shift to the fiber mode. The phase shift is proportional to the fiber tilts and the mode width/height.

Note that if the end surface is at the receiving fiber location (typically near focus) then a tilt of the fiber introduces phase tilt. If the end surface is far from the fiber, then tilt of the fiber introduces a decenter of the mode. Therefore, great care must be used in selecting the tilt and decenter values; proper use of these values depends upon knowledge of where the overlap integral is being computed (at the end surface!) relative to the fiber location.

Choosing the location for the receiving fiber

The receiving fiber mode may be either centered on the incoming beam chief ray or at the end surface vertex. In this latter case, the fiber mode is decentered by the amount equal to the incoming chief ray's x and y coordinates on the end surface, and the fiber is tilted to be aligned with the local Z axis rather than the chief ray.

Quantitative beam analysis

There are many different ways to analyze a beam. Zemax can compute total power, peak irradiance, encircled energy, centroid locations, beam width, M-squared values, and various RMS values. Because of the large number of data that can be computed, some of the data values are only available via the POPD optimization operand; for details, search the help files for "POPD". The expressions in the following sections are explicit in the x direction coordinate, but similar expressions for y of course exist.

Beam coordinates and pilot beam properties

Most basic beam data is displayed on the POP feature output window, and is also available from the POPD operand.

Peak Irradiance and total power

The peak irradiance is the maximum power per unit area at any point in the beam. Irradiance is measured in Source Units per lens unit squared. For details, search the System Explorer section of the Setup Tab help file for "Units". The total power is the integral of the irradiance over the entire beam.

Centroid locations

The beam centroid is given by:

$$c_x = \frac{\int x\, I(x, y)\, dx\, dy}{\int I(x, y)\, dx\, dy},$$

where I(x, y) is the irradiance of the beam.

Beam width and M-squared

The following method and theoretical basis for defining beam width and M-squared is from A. E. Siegman, and is applicable to all beams, Gaussian or not, coherent and incoherent.

The second moment of an arbitrary beam is defined as the square root of:

$$\sigma^2{}_x = \frac{\int (x - c_x)^2\, I(x, y)\, dx\, dy}{\int I(x, y)\, dx\, dy},$$

where c_x is the centroid coordinate as defined earlier. The beam width (defined here as the beam radius at the $1/e^2$ intensity point) is then $W_x = 2\sigma_x$, or twice the second moment. Note the beam width will change as the beam propagates, and is thus a function of the beam z coordinate relative to the waist. The smallest value for the beam size is $W_x(0)$. The beam width at other locations is given by

$$W_x^2(z) = W_x^2(0) + M_x^4 \left(\frac{\lambda z}{\pi W_x(0)} \right)^2,$$

where M_x is a parameter characteristic of the beam. This expression allows definition of the desired M-squared factor as

$$M_x^2 \cong \left(\frac{\pi}{\lambda z} \right) W_x(0) W_x(z), \text{ for } z \to \infty.$$

Wavefront error and RMS beam deviations

The mean, RMS, or peak-to-valley (PTV) irradiance or phase variation may be computed for any beam by using the appropriate settings for the POPD operand.

Encircled energy

Encircled energy is the fraction of the total power or energy contained with a circle of a specified radius r, centered on the chief ray reference point, the beam centroid, or a surface vertex reference. This fraction is always between zero and 1 as is defined as:

$$f = \frac{\int_o^r I(r)\, dr}{\int_o^\infty I(r)\, dr}.$$

A variation on encircled energy is to compute the radial coordinate at which a specified fraction is achieved.

Suggestions for use

It frequently takes some experimentation with the sampling, width, and surface specific settings (see above)

to get good, accurate results from this feature. Keep in mind the following suggestions:

Use an appropriate width for the initial beam. The best initial width to use is almost always that computed by the "Auto" button next to the width control, and the use of this control is highly recommended. Note that if the width is too large, then there will not be enough points across the beam to yield adequate sampling, if the width is too small, aliasing will occur. If the beam waist is smaller than the wavelength at any surface, including the initial beam, the POP results are probably not accurate, see "Algorithm assumptions" .

> **Always check the initial beam by setting the end surface to the start surface in the analysis window, and observe that the beam is properly sized for the selected array width.**

Use adequate sampling. As the guard beam increases in size relative to the non-zero amplitude portion of the beam, the number of points that have non-zero data will decrease. It may be required to increase the sampling to be sure enough points sample the beam.

If there are any cylindrical or toroidal optics in the system, or if the beam itself is astigmatic or anamorphic, try using "Separate X, Y" on the beam definition tab. Separating the X and Y propagation greatly reduces the sampling required for accurate results because a reference bispherical surface is used instead of a reference sphere, and the bisphere is capable of more closely fitting anamorphic beams.

If the results seem unbelievable, debug the propagation by propagating one surface at a time through the optical system, and study the results for plausibility. Typically this process will highlight at which surface the algorithm fails. It is frequently the case that certain surfaces, such as gradient index optics or highly tilted surfaces, are best handled by ray tracing. For these surfaces, choose the "Use Rays To Propagate To Next Surface" option.

Imagine a large diameter beam with a wavelength of 1.0 micrometers (0.001 mm), an array width of 64 mm, and a sampling of 64 x 64 points. The delta between points is 1.0 mm. Now imagine the beam going through a collimating lens with a focal length of 100.0 mm. The sampling spacing near focus is given by:

$$\Delta x_2 = \frac{\lambda |z_2|}{n_x \Delta x_1},$$

which yields a new sampling spacing of 0.0015625 and a total width of 0.1 mm. To decrease the sample spacing in the focal plane while keeping the array width constant, the product $n_x \Delta x_1$ must increase. Since the sample spacing is the array width divided by the number of points, both the number of points and the initial array width would have to be doubled. If the aberrations are so large that the beam size is on the order of the array width in the focal plane, then the array width must increase. Note the array width in the focal plane is given by:

$$n_x \Delta x_2 = \frac{\lambda |z_2|}{\Delta x_1}.$$

Increasing the array width can be accomplished by increasing the number of points while leaving the initial array width fixed.

> **Note that the beam sampling and width may be manually adjusted at any surface.**

See "Surface specific settings" for information on resizing the beam at any surface.

Algorithm assumptions

There are numerous approximations made in the model and algorithms used for physical optics modeling. Some of these assumptions will limit the accuracy of the results in certain cases. The major assumptions are described below.

Both the angular spectrum and the Fresnel diffraction propagation algorithms are developed assuming the beam is not too fast. Neither theory accurately predicts the correct diffraction results if the F/# of the beam is too low. The errors in the propagation theory will reveal themselves as discontinuities in the data as the propagation algorithm crosses over the Rayleigh range boundary. The loss of accuracy cannot be precisely quantified (if Zemax could precisely quantify the error, we could apply that as a correction to have a more accurate theory). Zemax will report a warning in the propagation report if the Gaussian pilot beam waist becomes smaller than the wavelength. This is a good rule to use as a limit of the accuracy of POP in general. Gaussian beams whose waist is small compared to the wavelength, such as fibers, act essentially as point sources radiating into a sphere. In this domain, the ray model is perfectly acceptable. In other words, don't use POP for very fast beams! The big differences between ray based and POP based results show up in very slow diffracting beams, not fast beams, so use ray based computations in Zemax to model very fast beams with confidence. Specifically, use the ray-based fiber coupling algorithm rather than POP for fast, accurate results in systems where the beam is fast and the diffraction effects from the edges of the lenses are not significant; this applies to virtually all fiber coupling systems.

Scalar diffraction theory applies. The vector nature of the electric field is ignored. For very fast beams, scalar diffraction theory is not valid and the results should be used with caution. Note that propagation of fast beams is generally well handled by geometric ray tracing, except near focus. Although there is no exact cut-off to the accuracy of scalar diffraction theory, beams faster than F/1 are probably not accurately modeled with scalar theory. This limitation is another reason not to use POP for very fast beams, as the previous paragraph explains.

Diffraction effects of surface structures are ignored. Gratings, binary optics, and diamond turned surfaces have a complex micro surface structure that can significantly effect the local electric field. This may or may not affect the propagation of the beam. Note that ray tracing, which also ignores the surface structure, has proved to be very useful and accurate for modeling these devices. Any surface adequately modeled by ray tracing will also work fine with physical optics propagation, since ray tracing is used to compute the surface transfer function.

The POP algorithm generally separates the electric field into two beam local orthogonal directions. The pilot beam properties, which are used to reference the phase of the electric field, are also computed along these two orthogonal directions. If the beam is incident upon an optical surface where the plane of incidence is not aligned with these projections, and if the optical surface has diffractive power that is also not aligned with these projections, the POP algorithm cannot accurately predict the correct electric field after diffraction, and the predicted beam will appear to rotate slightly. The error is usually only noticeable for beams that are incident at large skew angles to very steeply tilted diffraction gratings that have significant diffractive bending of the beam (that is, a diffractive order of 0 would describe a significantly different ray path than the diffractive order being used would). These types of systems are well modeled by conventional ray tracing and POP should not be used if the beam rotation is apparent.

When performing polarized beam analysis, the input and fiber mode beam (if any) must contain polarized electric field data. If a ZBF file or DLL is used to define the input or fiber mode beam, and no polarized data is contained in the file or created by the DLL, and a polarized beam analysis is requested, Zemax will create a polarized beam from the unpolarized data. The method used is to set the two orthogonally polarized electric fields equal in amplitude and phase, normalizing to conserve the total power in the beam. This allows polarized beam analysis, including transmission through coatings, interfaces, and bulk media.

Samples

Many sample files are included with Zemax to demonstrate some of the applications and proper use of the physical optics modeling features. The files are located in the <data>\Samples\Physical Optics folder. Some of these files are described below. Because of the large amount of computation involved in physical optics propagation, some of these files take some time to compute and display the results.

Free space propagation

The sample file "Basic Propagation" illustrates the case of a beam propagating through free space. The beam is a Gaussian defined to have a waist of 0.1 at surface 1. The wavelength is chosen to be π micrometers so that the Rayleigh range is 10.0 mm. At a distance of 1 Rayleigh range, the beam expands in size by $\sqrt{2}$, and the peak irradiance drops to 0.5. At a distance of 2 Rayleigh ranges, the peak irradiance drops to 0.2, and at 3 Rayleigh ranges the irradiance decreases to 0.1.

Note that beam may be virtually propagated backward using the normal Zemax sign convention of a negative thickness. Choosing the "End Surface" to be any of the surfaces in the LDE will show the beam intensity at that surface. Note the phase of the beam along the axis has the correct Gouy shift.

A pinhole aperture

The sample file "Pinhole Aperture" illustrates spatial filtering. The first lens forms an aberrated image, which can be seen by selecting the end surface to be 5. Surface 6 has a small circular aperture, which only allows the central portion of the beam to pass. The second lens recollimates the beam. The spatially filtered beam can be seen at the image of the stop on surface 10. There are several interesting things to note:

The transmitted irradiance drops to 0.12. The smaller the pinhole, the lower the transmitted irradiance. Because the pinhole aperture changes the properties of the beam, the pilot beam parameters must be recomputed. Note the option to do so is selected on surface 6.

A lens array

The sample file "Lenslet Array" uses the user defined surface DLL "US_ARRAY.DLL" to create a 7 x 7 array of rectangular lenses. The focal distance is 100.0 mm, and the pilot beam will focus down to a waist size of about

10 micrometers. However, the lenslet array forms multiple focal spots. To see all of the spots requires high sampling so that all the spots can be seen without aliasing. Notice the decrease in intensity and the aberrations apparent in the spots towards the outside. It is also possible to see the rectangular structure in the diffracted image spots due to the rectangular apertures of the lenslets.

Talbot imaging

Talbot imaging refers to the property of a beam forming an image of itself as it propagates. The image reveals itself in both phase and amplitude modulations as the beam propagates. The sample file "Talbot Imaging" shows a user defined aperture on surface 1 that consists of 20 rectangular slits, like a grating. A propagation of 20 mm reveals the phase reversed Talbot image of the grating. An additional 20.0 mm shows the restored image of the grating. The image is not perfect because of the finite extent of the grating. Note the region near the center of the grating has a well formed image.

Fresnel lens

The sample file "Fresnel Zone Plate Lens" illustrates the focusing power of an aperture consisting of concentric rings. The spacing of the rings is selected to block light from every other Fresnel zone, so that at a distance of

200.0 mm the light diffracted from all the zones interferes constructively. The constructive interference creates a bright focused spot on axis; an effect not predicted by ray tracing. This file also illustrates the intensity of the beam at several planes between the Fresnel zone plate and focus; note the concentric ripples caused by interference from the edges of the zone plate.

Since Zemax does not have a multiple-zoned circular aperture, the zone plate was created by placing one annular obscuration on each of several surfaces; all located at the same plane. The option to "Use Rays To Propagate To Next Surface" is used to speed the computation through all the aperture surfaces.

Sequential Surfaces

Zemax models many types of optical components. These include conventional spherical glass surfaces, plus aspheres, toroids, cylinders, and others, Zemax can also model components such as diffraction gratings, binary optics, Fresnel lenses, holograms, and others.

Because of the large number of surface types Zemax supports, a conventional spreadsheet arrangement for the user interface would be difficult to use. For example, there is no need to have a column for diffraction order for a surface that has no diffractive capability. To make the user interface as uncluttered as possible, Zemax uses different surface types to indicate what kinds of data are needed to define that type of surface.

Parameter data for sequential surfaces

A Standard surface can be a plane, spherical, or conic aspheric surface which is followed by a homogeneous material (such as air, mirror, or glass). The only parameters required are a radius (which may be infinity to yield a plane), a thickness, a conic constant (the default zero value indicates a sphere), and the name of the glass type.

Other surface types use these same basic data, as well as other values. For example, the Even Asphere surface uses all the Standard surface column data plus additional values which describe coefficients on a polynomial. These values are called parameters. The most important property of the parameter values to understand is that their meaning changes depending upon the surface type selected. For example, the Even Asphere surface uses parameter 1 to specify the coefficient on the parabolic aspheric term. However, the Paraxial surface uses parameter 1 to specify the focal length of the surface. Both surface types use parameter 1, but for different purposes, since both surface types are never used on the same surface at the same time.

This sharing of data storage simplifies the Zemax interface, as well as reduces the total memory required to run the program. After you change a surface from Standard to some other surface type, Zemax will automatically change the column headings on the parameter columns to reflect what each parameter does on that surface. As you move the cursor from cell to cell, the column headings will always show you what that cell is used for. If the current surface does not use the parameter column, the column heading will display "Unused". For more information see "Lens Data".

Extra data for sequential surfaces

Some surfaces cannot be described by only a few parameter values. For example, the Binary Optic 1 surface type requires nine parameters as well as up to several hundred additional numbers. This extra data is displayed in the Lens Data Editor, starting with parameter 13 and up. (Note that previous versions of Zemax had a separate editor to hold this information, called the Extra Data Editor.) For the surface Binary Optic 1, parameter 13 specifies the "Maximum Term #", or the number of extra data terms available in the Lens Data Editor. Changing the "Maximum Term #" automatically adds or removes columns in the Lens Data Editor.

The screenshot below is of the default new lens file, with a surface changed to Binary Optic 1. The "Maximum Term #" is set to 5, so Zemax created 5 more columns for the extra data values: Norm Radius, X1Y0, X0Y1, X2Y0, X1Y1, and X0Y2.
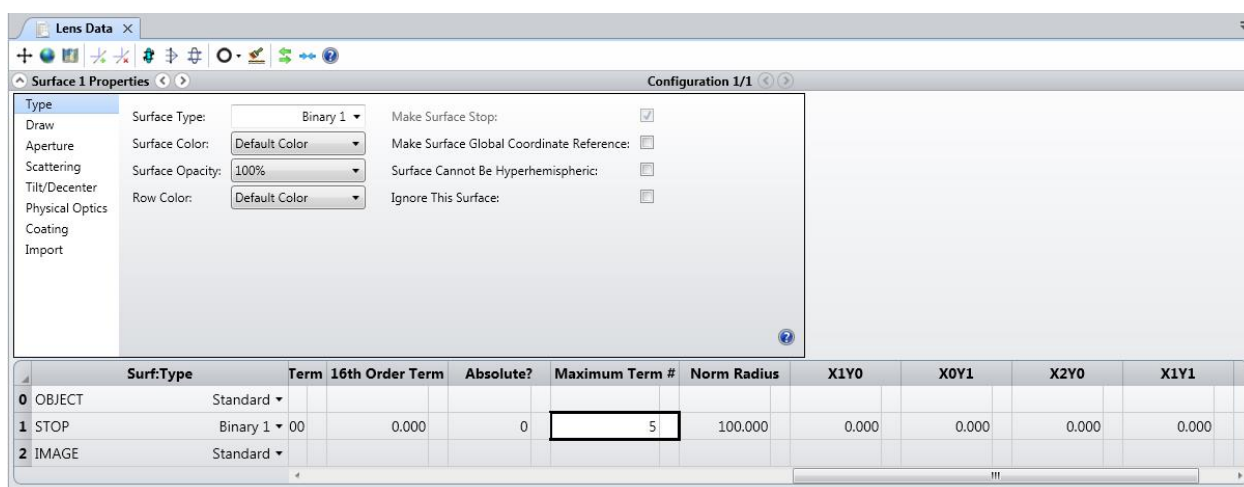
Highlighting a different surface shows the parameter number, as shown in the screenshot below. Binary Optic 1 uses the main lens data parameters 0 – 9, and the extra data values start with parameter 13. Note that because Binary Optic 1 does not use the default lens data parameters 10 – 12, they are not listed.
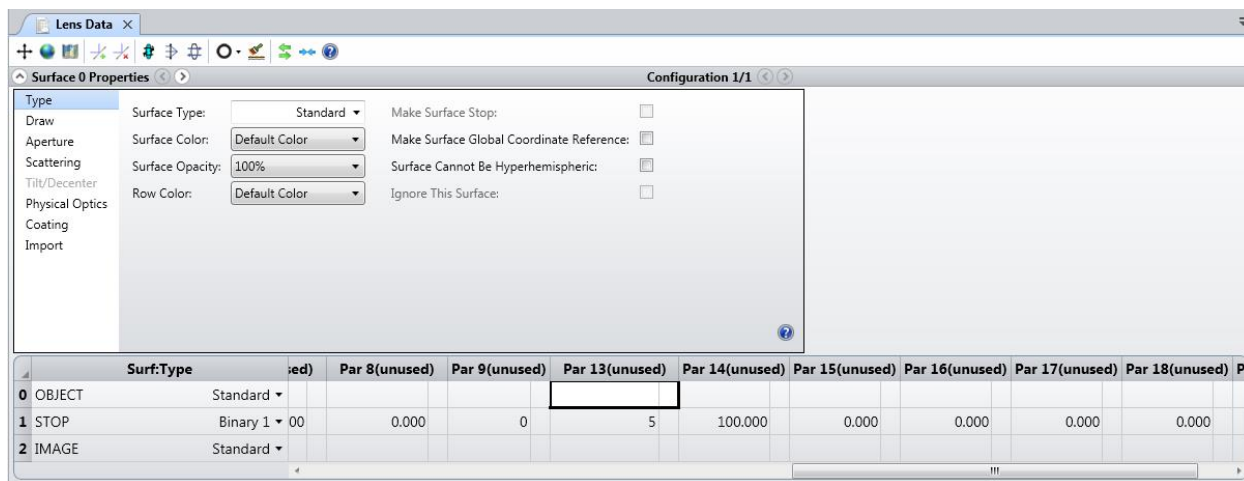
If a surface uses extra data values, the parameters and their definitions are listed under the surface description in the "Surface types" section.

These extra data value may be edited just like any other value in the Lens Data Editor. Alternatively, extra data may be imported into the Lens Data Editor via the "Import" section of the Surface Properties dialog. The extra data values are imported from data files which end in the extension .DAT. See the Lens Data Editor for more information.

Sequential Surface Types

Zemax models planes, spheres, and conics; all of these surface types are grouped under the category of Standard surface. By double clicking on the Surface Type column other surface models can be selected. A pop- up dialog will appear which lists all of the available surface types which can be selected. Zemax supports many different types of surfaces in addition to the Standard surface.

User defined sequential surfaces

No matter how many surfaces are added to Zemax, there always seems to be the need to add another surface type to solve a particular design, modeling, or tolerancing problem. If the surface type needed for a problem is not already included with Zemax, it is fairly easy to add new surface types using the User Defined surface described in "User Defined". User defined surfaces are created by writing software that defines the properties about the surface, and then dynamically linking the software into Zemax.

If you have a need for a custom surface type, and you do not wish to write the software for it yourself, please contact Zemax Technical Support for a quote on developing a custom surface to meet your requirements. We have considerable experience in developing ray tracing algorithms, and can generally write user defined surface code for a small fee, often on very short notice.

Summary table of sequential surface types

The types of surfaces supported by Zemax are summarized in the following table. The most common surface type is the Standard surface. For types not listed, see "User Defined".

SUMMARY OF SURFACE TYPES

Surface Model	Description
ABCD	Uses ABCD matrix to model a "black box".
Alternate Even	Even Asphere surface with alternate solution selected.

Alternate Odd	Odd Asphere surface with alternate solution selected.
Atmospheric Refraction	Refraction caused by looking through Earth's atmosphere.
Biconic	A conic asphere in X and Y independently.
Biconic Zernike	A biconic surface with x, y, and Zernike polynomial terms added.
Binary Optic 1	Uses 230 term polynomial to define phase.
Binary Optic 2	Uses radial polynomial to define phase.
Binary Optic 3	Dual zone aspheric and diffractive surface.
Binary Optic 4	Multiple zone aspheric and diffractive surface.
Birefringent In/Out	For modeling uniaxial crystals; supports tracing of rays in ordinary or extraordinary mode. Birefringent gradient materials with variable crystal axis orientation are also supported.
Black Box Lens	A surface that models a range of Zemax surfaces, but the defining data is hidden from the user.
Conjugate	Defines surface with perfect imaging at two points.
Coordinate Break	Permits rotation and decentration.
Cubic Spline	Rotationally symmetric fit to 8 points.
Cylindrical Fresnel	Polynomial cylindrical Fresnel on a polynomial cylindrical surface.
Data	Dummy surface that passes extra data values to a UDS.
Diffraction Grating	Ruled grating on Standard surface.
Elliptical Grating 1	Elliptical grating with aspheric terms and polynomial grooves.
Elliptical Grating 2	Elliptical grating with aspheric terms and grooves formed by tilted planes.
Even Asphere	Standard surface plus polynomial asphere terms.
Extended Asphere	Uses radial polynomial to define sag.
Extended Cubic Spline	Rotationally symmetric fit of up to 250 points.
Extended Fresnel	Polynomial Fresnel on a polynomial surface.
Extended Odd Asphere	Uses odd terms of radial powers.
Extended Polynomial	Uses 230 term polynomial expansion to define sag.

Extended Toroidal Grating	An aspheric toroidal grating with extended polynomial terms
Filter	A surface to filter or apodize the beam with an arbitrary function.
Fresnel	Plane surface with refractive power.
Generalized Fresnel	XY polynomial Fresnel on an aspheric substrate.
Gradient 1	Radial gradient index material surface.
Gradient 2	Radial gradient index material surface.
Gradient 3	Radial and axial gradient index material surface.
Gradient 4	X, Y, and Z gradient index material surface.
Gradient 5	Radial and axial gradient index material surface with dispersion model.
Gradient 6	Radial gradient index material surface with Gradient Lens Corp. dispersion model.
Gradient 7	Spherical gradient profile.
Gradium™	Axial gradient index material surface with dispersion model.
Gradient 9	Radial gradient index material surface with NSG SELFOC lens dispersion model.
Gradient 10	Y gradient index material surface with dispersion model.
Grid Gradient	A gradient index surface described by a 3D grid of points.
Grid Phase	A phase surface described by a grid of points.
Grid Sag	A surface shape described by a grid of points.
Hologram 1	Two-point optically fabricated hologram.
Hologram 2	Two-point optically fabricated hologram.
Irregular	A Standard surface with decenter, tilt and other deformations.
Jones Matrix	General Jones Matrix for modifying polarization state.
Lenslet Array	An array of lenslets.
Non-Sequential Components	For tracing rays non-sequentially through a collection of 3D surfaces and objects.
Odd Asphere	Standard surface plus polynomial asphere terms.

Odd Cosine	Odd Asphere surface plus cosine polynomial terms.
Optically Fabricated Hologram	Optically fabricated hologram with arbitrary construction optics and elliptical substrate.
Paraxial	Thin lens surface, has ideal behavior.
Paraxial XY	Thin lens with separate specification in X,Y.
Periodic	Cosine shaped surface.
Polynomial	Polynomial expansion in x and y.
Q-Type Asphere	An asphere based upon the Forbes polynomials
Radial Grating	A diffraction grating with radial phase profile
Radial NURBS	Uses NURBS curve to define a rotationally symmetric surface.
Retro Reflect	Retro reflects rays back along incident path.
Slide	A plane surface that allows a bitmap image to act as a filter.
Standard	Includes planes, spheres, and conics.
Superconic	Superconic asphere with fast convergence.
Tilted	Defines a tilted surface without changing coordinate systems.
Toroidal	Models conic and aspheric toroids and cylinders, with added Zernike terms.
Toroidal Grating	Ruled grating on a conic toroid.
Toroidal Hologram	Toroidal substrate with two point optically fabricated hologram.
Toroidal NURBS	Uses NURBS curve to define a toroidally symmetric surface.
User Defined	A general surface which uses an arbitrary user defined function to describe the refractive, reflective, diffractive, transmissive, or gradient properties of the surface.
Variable Line Space Grating	Variable line space grating surface.
Zernike Fringe Phase	Uses 37 Zernike Fringe polynomials to define phase.
Zernike Fringe Sag	Uses 37 Zernike Fringe polynomials to define sag.
Zernike Standard Phase	Uses up to 231 Zernike Standard polynomials to define surface phase.
Zernike Standard Sag	Uses up to 231 Zernike Standard polynomials to define surface sag.
Zernike Annular Phase	Uses up to 231 Zernike annular polynomials to define surface phase.

| Zone Plate | Fresnel Zone Plate model using annular rings of varying depth. |

Sequential surface type definitions

The following is an alphabetical list of surface types, including their associated parameter definitions. The main parameters are numbered up to 12, and any extra data parameters start at 13. For more information see "Parameter data" and "Extra Data".

ABCD

The ABCD surface provides a powerful method for modeling "black box" optical systems. If you have a lens (or a complete optical system) which is only a subsection of what you wish to model, and you do not have prescription data for the individual components, you can still model the behavior to first order.

The ABCD surface accepts eight parameters: Ax, Bx, Cx, Dx, Ay, By, Cy, and Dy. These are used to form two- by-two matrices (one for the x-direction and one for the y) which are used to alter the ray as it crosses the surface. The exit ray is related to the incident ray by

$$
\begin{bmatrix} x' \\ \omega'_x \end{bmatrix} = \begin{bmatrix} A_x & B_x \\ C_x & D_x \end{bmatrix} \begin{bmatrix} x \\ \omega \end{bmatrix},
$$

with a similar expression for the y components. Because there is no reliable way to compute the phase through an ABCD surface, any calculation that requires OPD data, such as OPD fans, MTF, and Zernike coefficients, will not be supported if an ABCD surface is present in the lens description.

PARAMETER DEFINITIONS FOR ABCD SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Ax	Bx	Cx	Dx	Ay	By	Cy	Dy

Alternate Even

There are two solutions to the ray-surface intercept equations used in tracing a ray to the next optical surface, if that surface is of the form of the standard or even aspheric surface types. Zemax selects the correct solution in the vast majority of cases. However, in certain systems, so-called "strange" rays are actually intended to intersect the next surface at the other, "alternate" solution. Strange rays are most likely to occur after a grazing incidence reflection in which the ray is still traveling in the same direction (the Z component of the ray vector does not change sign). The alternate even surface model is identical to the even asphere surface model, except the alternate solution is used. Zemax may not be able to correctly compute the optical path difference when the alternate even surface is being used.

Alternate Odd

There are two solutions to the ray-surface intercept equations used in tracing a ray to the next optical surface, if that surface is of the form of the standard or even aspheric surface types. Zemax selects the correct solution in the vast majority of cases. However, in certain systems, so-called "strange" rays are actually intended to intersect the next surface at the other, "alternate" solution. Strange rays are most likely to occur after a grazing incidence reflection in which the ray is still traveling in the same direction (the Z component of the ray vector does not change sign). The alternate odd surface model is identical to the odd asphere surface model, except the alternate solution is used. Zemax may not be able to correctly compute the optical path difference when the alternate odd surface is being used.

Atmospheric Refraction

This surface is used to simulate the effects of refraction through the Earth's atmosphere when viewing a star or point source. The atmosphere has a small but non-zero dispersion which introduces a tilt term to the incoming wavefront which depends upon wavelength. Zemax uses a model based upon those described in the following publications:

P. K. Seidelmann, Ed., "Refraction - Numerical Integration", Section 3.281, Explanatory Supplement to the Astronomical Almanac, pp. 141-143, University Science Books, Mill Valley, 1992.

C. Y. Hohenkerk and A. T. Sinclair, NAO Technical Note 63, Royal Greenwich Observatory Science and Engineering Research Council, 1985.

Six parameters are supplied to the model: The observed zenith angle of the source in degrees, the height of the observer above sea level in meters, the ambient temperature at the observer in Kelvin, the atmospheric pressure at the observer in millibars, the relative humidity (a number between 0.0 and 1.0), and the latitude of the observer in degrees.

Zemax computes the atmospheric refraction angle in radians for all defined wavelengths, then subtracts from all wavelengths the amount of refraction at the primary wavelength; thereby using the primary wavelength as a reference. To disable the referencing to the primary wavelength, set the "Absolute ?" flag to 1. Atmospheric refraction manifests itself as a small tilt in the OPD fan plot, or a slight chief ray offset similar to lateral color in the ray fan plot. All refraction is assumed to occur in the y direction only.

This surface models color separation. To model atmospheric turbulence, use the Grid Phase surface with an externally provided data file. For information on the Grid Phase surface, see "Grid Phase".

PARAMETER DEFINITIONS FOR ATMOSPHERIC REFRACTION SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7
Zenith	Height	Temperature	Pressure	Humidity	Latitude	Absolute ?

Biconic

The biconic surface is similar to a toroidal surface, except the conic constant and base radius may be different in the X and Y directions. The biconic surface allows specification of Rx, Ry, Kx, and Ky directly. The sag of a biconic is given by

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x^2 x^2 - (1 + k_y)c_y^2 y^2}}, \quad \text{where} \quad c_x = \frac{1}{R_x}, \, c_y = \frac{1}{R_y}.$$

The radius in the x direction is set in the parameter 1 column. If set to zero, the x radius is interpreted to be infinity.

PARAMETER DEFINITIONS FOR BICONIC SURFACES

Parameter 1	Parameter 2
R_x	k_x

Biconic Zernike

The biconic Zernike surface is similar to a biconic surface, with the added capability to add X, Y, and Zernike polynomial deformations. The sag of a biconic Zernike is given by:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x^2 x^2 - (1 + k_y)c_y^2 y^2}} + \sum_{i=1}^{16} \alpha_i x^i + \sum_{i=1}^{16} \beta_i y^i + \sum_{i=1}^{N} A_i Z_i(\rho, \varphi),$$

$$\text{where} \quad c_x = \frac{1}{R_x}, \, c_y = \frac{1}{R_y},$$

and the Zi terms are the Zernike standard terms as described in "Zernike Standard Sag".

Parameter 0 is the extrapolate flag. If the extrapolate flag is set to 0, the Zernike terms are ignored outside of the normalization radius. If the extrapolate flag is set to 1, then the Zernike terms are considered no matter where the ray lands on the surface; even if the ray lands beyond the normalization radius. The radius and conic in the x direction are set in the parameter 1 and 2 columns. If the radius is set to zero, the x radius is interpreted to be infinity. Parameter 3 is the maximum number of Zernike terms, this value may be between 0 and 210, inclusive. Parameter 4 is the normalization radius used in the Zernike expansion.

PARAMETER DEFINITIONS FOR BICONIC ZERNIKE SURFACES

Parameter 0	Parameter 1	Parameter 2	Parameter 3	Parameter 4
Extrapolate	R_x	k_x	N	Normalization Radius

EXTRA DATA DEFINITIONS FOR BICONIC ZERNIKE SURFACES

Parameter Number	Description
5-12	Unused
13-28	Coefficients on powers of x
29-44	Coefficients on powers of y
45-254	Coefficient on the Zernike Standard polynomials

Binary Optic 1

Binary optics, also called kinoforms, are similar to holograms and diffraction gratings in that small grooves or lines across the optical surface impart a change in phase of the wavefront passing through the surface. Zemax does not model the wavelength-scale grooves directly. Instead, Zemax uses the phase advance or delay represented by the surface locally to change the direction of propagation of the ray. Other effects, such as efficiency or multiple order diffraction are ignored. Binary surfaces can have zero thickness, with no index change across the surface, or may have different media on either side of the surface. In this case, the refraction due to the material change will be accounted for as well as the diffraction effects. Full exact polychromatic ray and OPD tracing is supported.

The binary optic 1 surface is similar to the extended polynomial surface, except the polynomial terms represent the variation in phase (rather than surface height) across the optic surface. The coefficients therefore have units of radians rather than lens units. The shape of the binary optic 1 surface is identical to the even asphere surface type; planes, spheres, conics, and polynomial aspheres up to order 16 are supported. The sag of the surface is given by the following expression:

$$ z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}, $$

where the terms are identical to those in the even asphere surface model. See "Even Asphere" for a complete discussion. The binary optic 1 surface adds phase to the ray according to the following polynomial expansion:

$$ \Phi = M \sum_{i=1}^{N} A_i E_i(x, y) $$

where N is the number of polynomial coefficients in the series, Ai is the coefficient on the ith extended polynomial term, and M is the diffraction order. The polynomials are a power series in x and y, as described in "Extended Polynomial". The coefficients Ai all have units of radians (2π radians is one wave). If Parameter 9 is any value other than zero, then the absolute value of the x and y coordinates are used for computing the phase in the above expression. This "Absolute" option yields a different, generally discontinuous phase profile.

PARAMETER DEFINITIONS FOR BINARY OPTIC 1 SURFACES

Param 0	Param 1-8	Param 9
M, the diffraction order	α_1 - α_8	Absolute? (see discussion)

EXTRA DATA DEFINITIONS FOR BINARY OPTIC 1 SURFACES

Parameter Number	Description
10-12	Unused
13	Maximum term number.
14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation
15-244	Polynomial terms
245 and above	Not used

The "Maximum term number" is used to specify the maximum polynomial term to be used in calculating the surface phase. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored. For example, if the last term in the series you wish to use is xy^3, which is term number 13, then specify "13" in the maximum term number column. Note that the term number is 13, because it is the 13th term in the polynomial expansion (not the 13th parameter in the editor).

The normalization radius scales the X and Y intercept coordinates of the ray so that the polynomial terms are all dimensionless and the coefficients are all in radians. For modeling phase surfaces that are not well described by polynomials, see "Grid Phase".

Binary optic coefficients sign conventions

It is important to understand the binary optic surface phase coefficients sign convention. Imagine a collimated beam being focused by an infinitesimal thin binary optic surface. If the collimated beam becomes a focused, converging beam on the other side of the binary optic, then the path length of a marginal ray is longer than the path length of the axis ray. Therefore, the binary optic must add a negative path length to the marginal ray. Using this convention, a positive power binary optic has a negative quadratic phase coefficient. Although the sign convention chosen by Zemax is arbitrary, it may be important to know what the convention is as far as fabrication decisions are concerned. It is always a good idea to test a few cases to verify the sign convention being used by the software before committing a design to fabrication.

Binary Optic 2

Binary optics are similar to holograms and diffraction gratings in that small grooves or lines across the optical surface impart a change in phase of the wavefront passing through the surface. Zemax does not model the wavelength-scale grooves directly. Instead, Zemax uses the phase advance or delay represented by the surface locally to change the direction of propagation of the ray. Other effects, such as scattering, efficiency, or multiple order diffraction are ignored. Binary surfaces can have zero thickness, with no index change across the surface, or may have different media on either side of the surface. In this case, the refraction due to the material change will be accounted for as well as the diffraction effects. Full exact polychromatic ray and OPD tracing is supported.

The binary optic 2 surface is similar to the extended asphere surface, except there are additional polynomial terms which represent the variation in phase (rather than surface height) across the optic surface. The coefficients therefore have units of radians rather than lens units. The shape of the binary optic 2 surface is identical to the even asphere surface type; planes, spheres, conics, and polynomial aspheres up to order 16 are supported. The sag of the surface is given by the following expression:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16},$$

where the terms are identical to those in the Even Asphere Surface model. See "Even Asphere" for a complete discussion. The Binary Optic 2 surface adds phase to the ray according to the following polynomial expansion:

$$\Phi = M \sum_{i=1}^{N} A_i \rho^{2i}$$

where N is the number of polynomial coefficients in the series, A_i is the coefficient on the 2ith power of ρ, which is the normalized radial aperture coordinate, and M is the diffraction order. The first extra data term is N, the number of terms, which can be zero to exclude all binary effects, or any integer up to 240 (or the ρ^{480} term). Extra data value number 2 is the normalization radius. Extra data values 3 through 242 are the coefficients.

PARAMETER DEFINITIONS FOR BINARY OPTIC 2 SURFACES

Param 0	Param 1	Param 2	Param 3	Param 4	Param 5	Param 6	Param 7	Param 8
M	α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8

EXTRA DATA DEFINITIONS FOR BINARY OPTIC 2 SURFACES

Parameter Number	Description
9-12	Unused
13	Maximum term number.

14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation
15	Coefficient in radians on ρ^2
16	Coefficient in radians on ρ^4
n	Coefficient in radians on $\rho^{2(n-14)}$

The "Maximum term number" is used to specify the maximum polynomial term to be used in calculating the surface phase. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored. For example, if the last term in the series you wish to use is ρ^{14}, which is polynomial term number 7, then specify "7" in the maximum term number column. Note that the term number is 7, because it is the 7th term in the polynomial expansion (not the 7th parameter in the editor).

The normalization radius scales the X and Y intercept coordinates of the ray so that the polynomial terms are all dimensionless and the coefficients are all in radians. For modeling phase surfaces that are not well described by polynomials, see "Grid Phase".

Binary optic coefficients sign conventions

See "Binary Optic 1" for a discussion of sign conventions.

Binary Optic 3

The Binary Optic 3 surface is very similar to the Binary Optic 2 surface. The key difference is that the Binary Optic 3 supports two concentric radial zones, with independent radius, conic, and polynomial aspheric deformation and diffractive phase data for each zone. The surface is divided into two zones by two radial coordinates, A_1, and A_2. The inner radial zone extends from the center of the surface to the radial coordinate A_1. The outer radial zone extends from A_1 outward. The radial coordinate A_2 is used for normalizing the phase coefficients in the outer zone, even though the surface may extend past the coordinate A_2. The outer zone is offset from the inner zone to make the surface sag continuous across the zone boundary, unless the optional "break" parameter is set to 1. Zemax requires that $0 < A_1 < A_2$.

The sag of the surface in the inner zone is given by the following expression:

$$ z_1 = \frac{c_1 r^2}{1 + \sqrt{1 - (1 + k_1)c_1^2 r^2}} + \sum_{i=1}^{N} \alpha_{1i} r^{2i}, \text{ for } r \leq A_1, $$

where N is the number of aspheric terms which may be set by the user as described below. The value for the inner zone curvature, c_1, is the reciprocal of the radius of curvature specified in the Lens Data Editor. The inner zone conic constant is also set in the Lens Data Editor, in the usual conic column. A similar expression with different coefficients and an offset value is used to define the sag of the surface in the outer zone:

$$z_2 = z_o + \frac{c_2 r^2}{1 + \sqrt{1 - (1 + k_2)c_2^2 r^2}} + \sum_{i=1}^{N} \alpha_{2i} r^{2i} \quad \text{,for } r > A_1,$$

where the term z_o is chosen to make the surface continuous across the boundary between the inner and outer zones at the radial coordinate A_1, or $z_o = z_1 (A_1) - z_2 (A_1)$ (the value of z_o is temporarily set to zero while evaluating z_2 for this calculation). The outer zone radius of curvature (from which the value c_2 is computed) and the outer zone conic are set in the parameter data as described below. For a flat outer zone radius, use zero.

Both the inner and outer zones have a diffractive phase profile, with independent coefficients. The phase of the inner zone is given by:

$$\Phi_1 = M_1 \sum_{i=1}^{N} \beta_{1i} \rho_1^{2i}, \text{ and } \rho_1 = \frac{r}{A_1},$$

where N is the number of polynomial coefficients in the series, β_{1i} is the coefficient on the $2_i{}^{th}$ power of ρ_1, which is the normalized radial aperture coordinate, and M1 is the diffraction order. A similar expression describes the phase for the outer zone:

$$\Phi_2 = \delta_o + M_2 \sum_{i=1}^{N} \beta_{2i} \rho_2^{2i}, \text{ where } \rho_2 = \frac{r}{A_2}.$$

The phase offset serves a similar purpose to the sag offset, and is defined by $\delta_o = \Phi_1(1) - \Phi_2(A_1/A_2)$ (the value of δ_o is temporarily set to zero while evaluating Ω_2 for this calculation). Note that the normalization radius A_2 is used only for defining the radial aperture to normalize the phase coefficients for the outer zone. The outer zone of the surface, and the associated phase profile, may extend beyond this value.

The dual zone nature of this surface creates a complication when computing the phase of the surface as the zone boundary is crossed. The phase offset value δ_o makes certain the phase is continuous across the zone boundary. This is desirable for design and analysis purposes, because phase jumps of hundreds of waves make interpretation and analysis difficult. However, the phase offset is artificial, and this must be accounted for in the actual design. For optimum imaging properties, the outer zone should be in phase with the inner zone. This condition is met when the inner and outer zones differ in phase by an integral number of wavelengths at the boundary, or more to the point, if $\delta_o = J2\pi$, where J is some arbitrary integer. This condition can usually be met by a small change in A_1, as long as there is some difference in the slope of the phase on either side of the boundary. To make this boundary condition simple to meet, Zemax computes $\sin \delta_o$ and places this value in parameter 7. The merit function boundary operand PMVA can then be used to target this value to be zero. Note this value is computed from the phase data, and should not be user defined or made variable.

The difference in sag between the inner and outer zones can be accounted for two ways. If the "Break?" parameter is set to 0, then the outer one is shifted along the local z axis so that the surface is continuous at the zone boundary. If "Break?" is set to 1, then the outer sag is defined by the sag equation with no offset, and in general a discontinuity in the sag will result. This discontinuity will cause a large discontinuity in the OPD computations that Zemax cannot subtract out.

PARAMETER DEFINITIONS FOR BINARY OPTIC 3 SURFACES

Param 1	Param 2	Param 3	Param 4	Param 5	Param 6	Param 7	Param 8
R_2	k_2	A_1	A_2	M_1	M_2	$\sin\delta_o$	Break?

EXTRA DATA DEFINITIONS FOR BINARY OPTIC 3 SURFACES

Parameter Number	Description
9-12	Unused
13	Maximum term number.
14, 15, 16, 17	The coefficients α_{11}, β_{11}, α_{21}, and β_{21} if N >= 1.
18, 19, 20, 21	The coefficients α_{12}, β_{12}, α_{22}, and β_{22} if N >= 2.
22, 23, 24, 25, etc.	The pattern continues in groups of 4 coefficients for N groups

Binary optic coefficients sign conventions

See "Binary Optic 1" for a discussion of sign conventions

Binary Optic 4

The Binary Optic 4 surface is similar to the Binary Optic 2 and 3 surfaces. The key difference is that the Binary Optic 4 supports a variable number of concentric radial zones, with independent radial size, radius, conic, diffraction order, polynomial aspheric deformation, and diffractive phase data for each zone. The number of zones, even asphere coefficients, and phase coefficients, are all user definable, up to the maximum allowed number of values in an extra data file (see the Import section of the Surface Properties). Each zone requires 4 values (aperture, radius, conic, and diffraction order) plus a variable number of even aspheric and binary phase coefficients. The total number of terms is $N_z*(4 + N_a + N_p)$ where N_z is the number of zones, N_a the number of asphere terms, and N_p is the number of phase terms. The total may not exceed 242 terms. Nz must be between 1 and 60; Na and Np must be between zero and 20.

The surface is divided into zones. The first zone extends from the vertex to the radial coordinate, A_1, the second zone from A_1 to A_2, and this continues through the last zone. The radial coordinates are also used for normalizing the aspheric and phase coefficients within their respective zones. Each zone is offset from the prior zone to make the surface sag continuous across the zone boundary. Zemax requires that the first zone have a radial aperture larger than zero, and each subsequent zone must be larger than the previous one.

The sag of the surface in zone j is given by the following expression:

$$z_j(r) = \frac{c_j r^2}{1 + \sqrt{1 - (1 + k_j)c_j^2 r^2}} + \sum_{i=1}^{Na} \alpha_{ji} p^{2i} + z_o.$$

Note c, k, and the α terms are all unique to each zone. The term zo is chosen to make the surface continuous across the boundary between the current and prior zone at the radial coordinate A_{j-1}, or

$$z_o = z_{j-1}(A_{j-1}) - z_j(A_j)$$

(the value of z_o is temporarily set to zero while evaluating z_j for this calculation). The normalized coordinate p is given by r / A_j.

All zones also may have a diffractive phase profile, with independent coefficients. The phase in zone j is given by:

$$\Phi_j = \delta_o + M_j \sum_{i=1}^{Np} \beta_{ji} \rho_j^{2i},$$

where N_p is the number of polynomial coefficients in the series, β_{1i} is the coefficient on the 2_i^{th} power of ρ_j, which is the normalized radial aperture coordinate, and M_i is the diffraction order. The phase offset serves a similar purpose to the sag offset, and is defined by $\delta_o = \Phi_{j-1}(1) - \Phi_j(A_{j-1}/A_j)$ (the value of δ_o is temporarily set to zero while evaluating Φ_j for this calculation). Note that the normalization radius for the outermost zone is used only for defining the radial aperture to normalize the coefficients for the outer zone. The outer zone of the surface, and the associated phase profile, may extend beyond this value.

The multiple zone nature of this surface creates a complication when computing the phase of the surface as the zone boundary is crossed. The phase offset value δ_o makes certain the phase is continuous across the zone boundary. This is desirable for design and analysis purposes, because phase jumps of hundreds of waves make interpretation and analysis difficult. However, the phase offset is artificial, and this must be accounted for in the actual design. For optimum imaging properties, the outer zone should be in phase with the inner zone. This condition is met when the inner and outer zones differ in phase by an integral number of wavelengths at the boundary, or more to the point, if $\delta_o = Q2\pi$, where Q is some arbitrary integer. This condition can usually be met by a small change in A_j, as long as there is some difference in the slope of the phase on either side of the boundary. To make this boundary condition simple to meet, Zemax computes sin δ_o for each of the zone boundaries and places the sum of the squares (SS) of this value in parameter 4. The merit function boundary operand PMVA can then be used to target this value to be zero. Note this value is computed from the phase data, and should not be user defined or made variable.

PARAMETER DEFINITIONS FOR BINARY OPTIC 4 SURFACES

Param 1	Param 2	Param 3	Param 4
Nz	Na	Np	SS $\sin\delta_o$

EXTRA DATA DEFINITIONS FOR BINARY OPTIC 4 SURFACES

Parameter Number	Description
5-12	Unused
13	The radial aperture of zone 1
14	The radius of curvature of zone 1.
15	The conic constant of zone 1.

16	The diffraction order of zone 1
Next Na	The aspheric sag coefficients for zone 1.
Next Np	The phase coefficients for zone 1
Remaining terms	The pattern continues in groups of 4+Na+Np coefficients for Nz groups

Binary optic coefficients sign conventions

See "Binary Optic 1" for a discussion of sign conventions.

Birefringent In and Birefringent Out

This feature is only available in the Professional and Premium editions of Zemax

This pair of surfaces models uniaxial crystals, such as calcite. These types of crystals are described by a crystal axis; which defines the axis of symmetry for the material, and two dispersion curves; one defining the "ordinary" index and the other the "extraordinary" index. Such materials refract rays differently depending upon the polarization state of the ray and the angle the ray makes with respect to the crystal axis. These different types of refraction yield two different possible refraction angles for any specific ray, and so they are called birefringent or double refraction materials. For details on birefringent materials, see Saleh and Teich, Fundamentals of Photonics, Wiley Interscience. Only a brief description of these materials and their ray tracing properties is given here. Note that when rays propagate close to parallel to the crystal axis, some crystal materials, such as quartz, may exhibit an effect called optical activity that Zemax does not model. Optical activity may be accounted for using the Jones Matrix surface, see "Jones Matrix". Birefringent In/Out surfaces may be planes, spheres, conic aspheres, or toroidal in shape.

Zemax can also model uniaxial crystals whose crystal axis orientation varies within the material. Several different polynomial models for the form of the variation are supported. For more information see "Defining the crystal axis for inhomogeneous materials".

Birefringent materials refract rays according to Snell's law, but the effective index of refraction in the media depends upon the input polarization state and the angle the refracted ray makes with respect to the crystal axis. Ordinary rays are refracted according to

$$n\sin\theta = n_o\sin\theta',$$

where the subscript "o" indicates the ordinary index. This is just Snell's law, of course. For extraordinary rays, the refraction law is

$$n\sin\theta = n(\theta_w)\sin\theta'.$$

This is also Snell's law, but the effective index of refraction in the birefringent material is a function of the angle θ_w, which is the angle between the crystal axis vector $\hat{a}$ and the refracted wave vector k. Additionally, the ray vector $\hat{s}$, which is the vector pointing in the direction of energy propagation (also called the Poynting vector), does does not follow the wave vector $\hat{k}$, but makes a small angle with respect to $\hat{k}$. In isotropic media $\hat{k}$ and $\hat{s}$ are the same, so for most optical design we keep track of $\hat{k}$; here we must consider the ray and the wave vector being different. The angle θ_w is defined by $\cos\theta_w = \hat{k} \bullet \hat{a}$ .The effective index of refraction is defined by

$$\left(\frac{1}{n(\theta_w)}\right)^2 = \left(\frac{\cos\theta_w}{n_o}\right)^2 + \left(\frac{\sin\theta_w}{n_e}\right)^2,$$

where n_o is the ordinary and n_e is the extraordinary index of refraction.

The angle α between $\hat{k}$ and $\hat{s}$ is defined by

$$\cos\alpha = \hat{k} \bullet \hat{s}, \text{ where}$$

$$\tan\alpha = \frac{(n_e^2 - n_o^2)\tan\theta_w}{n_e^2 + (n_o\tan\theta_w)^2},$$

and the vectors $\hat{k}$ and $\hat{s}$ and are both coplanar with the crystal axis vector $\hat{a}$. The wave vector $\hat{k}$ points along the normal to the wavefront, while $\hat{s}$ points along the direction of energy propagation. For ray tracing purposes, Zemax uses the components of $\hat{s}$ as the ray direction cosines.

Defining the substrate shape

Birefringent In/Out surfaces may be planes, spheres, conic aspheres, or toroids. The default substrate shape is the same as a Standard surface (see "Standard") defined by the radius of curvature and a conic constant. This standard substrate is called shape 0. The other substrate shape currently supported is shape 1, which is a surface similar to the Toroidal surface (see "Toroidal") except the aspheric coefficients are limited to 10th order and no Zernike terms are supported. The shape number is defined by parameter 6.

Defining the index

To define the ordinary index, Zemax uses the glass catalog and glass name in the usual way. For example, to use Calcite, define the surface to be a birefringent "in" type, and enter Calcite for the glass name. Zemax uses this name to compute the ordinary index at any defined wavelength.

To compute the extraordinary index, Zemax appends "-E" to the glass name, and then looks for this material in the glass catalog. For the case of Calcite, Zemax searches for "Calcite-E" in the glass catalog, and if found, uses that for the extraordinary index. If not found, an error message is issued. This technique allows definition of any material with any ordinary and any extraordinary index to be defined and used.

Most Zemax calculations, such as EFL, EPD, F/#, etc. use the ordinary index of refraction. The extraordinary index is only used to trace extraordinary rays. The Prescription report will list both the ordinary and the extraordinary index, and this data should be carefully checked before using this feature for any critical analysis.

Defining the crystal axis for homogeneous materials

For homogeneous materials, the crystal axis orientation is constant throughout the media. The crystal orientation is defined using parameter columns 2, 3, and 4 for the x-, y-, and z- direction cosines of the axis. For example, to define the crystal axis along the x axis, the vector values are (1, 0, 0). For the axis aligned to the surface vertex normal, which is along the z axis, the vector is (0, 0, 1). Note the coordinates are local to the "Birefringent In" surface and Zemax internally converts to unit vector components.

Defining the crystal axis for inhomogeneous materials

Birefringent materials whose crystal axis orientation varies as a function of position are supported. For a discussion of the method used to model these types of materials, see C. Jenkins, R. Bingham, K. Moore, and G. Love, "Ray equation for a spatially variable uniaxial crystal and its use in the optical design of liquid-crystal lenses", J. Opt. Soc. Am. A, Vol. 24, No. 7, pp2089-2096 (2007). The variation in the crystal axis orientation is defined using one of several polynomials. The selection of the polynomial used is called the Gradient Mode, and the Gradient Mode is defined by Extra Data Value 1. The supported Gradient Modes are defined in the following table.

GRADIENT MODE POLYNOMIALS

Gradient Mode	Definition of $\hat{a}$
0	Material is homogeneous, $\hat{a}$ is defined by parameters 2-4 on the Lens Data Editor.
1	$$a_x = \sum_{i=0}^{6} c_{xi} r^i, \; a_y = \sum_{i=0}^{6} c_{yi} r^i, \; a_z = \sqrt{1 - a_x^2 - a_y^2}.$$
2	$$a_z = 1 + \sum_{i=1}^{8} c_{zi} r^i, \; a_x = \frac{x}{r}\sqrt{1 - a_z^2}, \; a_y = \frac{y}{r}\sqrt{1 - a_z^2}.$$
3	$$\theta = \sum_{i=0}^{8} p_i r^i, \; a_z = \sin\theta, \; a_x = \frac{x}{r}\cos\theta, \; a_y = \frac{y}{r}\cos\theta. \text{ If } \theta = 0 \text{ then } \hat{a} = \{0, 1, 0\}.$$ The angle θ is measured in radians.
4	$$a_x = 0, \; a_y = \sum_{i=0}^{8} a_i y^i, \; a_z = \sqrt{1 - a_y^2}.$$
5	$$\theta = \sum_{i=0}^{8} p_i r^i, \; a_z = \cos\theta, \; a_x = \frac{x}{r}\sin\theta, \; a_y = \frac{y}{r}\sin\theta.$$ The angle θ is measured in radians.
6	$$\sin\theta = \sqrt{\frac{O^2(E^2 - (E - A\phi)^2)}{B(E - A\phi)^2}}, \text{ where } \phi \text{ is the polar angle and } a_x = 0, \; a_y = \sin\theta,$$ $a_z = \cos\theta$. The angle ϕ is measured in radians. This formula models a radial vortex phase profile, where O and E are nominally the ordinary and extraordinary indices and A and B are used to define the index gradient around the vortex.
7	$$a_x = \sum_{i=0}^{16} c_{xi} r^i, \; a_z, \; a_y = \sqrt{1 - a_x^2 - a_z^2}.$$
8	$$a_x = 0, \; a_y = \sum_{i=0}^{4}\sum_{j=0}^{4} c_{2j2i} r^{2j}\left(\frac{z - z_0}{z_n}\right)^{2i}, \; a_z = \sqrt{1 - a_y^2}.$$
-	Other equations may be added in the future based upon user suggestions.

After computing the component values of the $\hat{a}$ vector using the equations in the table, the vector is normalized to be a unit vector. Care should be taken to ensure the polynomial coefficients define valid, real values for the crystal axis vector, otherwise the behavior of the crystal is undefined. All polynomial coefficients are defined in an extra data file (see the Import section of the Surface Properties) in the columns following the Gradient Mode value.

When using a non-zero Gradient Mode, rays will generally take curved paths within the crystal. The reason is the continuously varying angle between the $\hat{k}$ and $\hat{a}$ vectors creates a varying effective index of refraction. Ray tracing must be done in a piece-wise fashion, and the accuracy of the ray trace depends upon the maximum size of the step allowed. The maximum step size allowed is called "Delta T" and is defined by Extra Data Value 2. The situation is very similar to that of gradient index glass. For more information see "Discussion on maximum step size for GRIN surfaces".

Determining which ray is traced

Zemax will trace the ordinary ray if the mode is set to 0 or 2. If the mode is set to 1 or 3, the extraordinary ray will be traced. The mode is parameter 1 in the lens data editor. In sequential mode, Zemax cannot trace both rays at once, however it is easy to create a multi-configuration lens with the mode set to 0 in configuration 1, and to 1 in configuration 2; this allows inspection of both possible paths as well as simultaneous optimization and layouts of the traced rays. The difference between modes 0 and 2 for ordinary rays and modes 1 and 3 for extraordinary rays is described in the next paragraph.

Accounting for phase rotation

To account for phase rotation when performing polarization ray tracing, there are two limiting cases to consider. Conceptually, the beam splits into an ordinary beam and an extraordinary beam with a small angle between the two beams. If the beams then propagate a large distance, the ordinary and extraordinary beams will separate and become two distinct beams. If however the propagation distance is short, the ordinary and extraordinary beams coherently interfere and recombine, resulting in a single beam with a (usually) rotated polarization vector. Conceptually this is like tracing both the ordinary and extraordinary rays, then coherently recombining them after propagation through the birefringent media. Which model is used to determine the transmission and polarization properties may be selected by the mode.

If the mode is 0 or 1, then only the ordinary or extraordinary part, respectively, is traced. These modes model the extraordinary and ordinary rays as two separate beams, and the fraction of ray energy in the part of the ray that is not traced is discarded. No polarization rotation will be modeled using mode 0 or 1. For transmission computations, both rays need to be traced separately and the total energy computed. It is difficult to compute the effects of phase rotation using mode 0 or 1.

Modes 2 and 3 assume that the angular deviation between the ordinary and extraordinary beam is so small that the region of interest can be approximated by a single beam. If the mode flag is 2, then the ordinary ray is traced, however the phase rotation due to the extraordinary ray is accounted for. If the mode is 3, the extraordinary ray is traced, and the phase rotation due to the ordinary ray is accounted for. For modes 2 and 3, no energy is discarded, and the polarization of the ray is properly rotated by the birefringent media.

To model birefringence with sequential rays, Zemax must assume that the ordinary and extraordinary beams will either overlap entirely or not at all. These limitations do not exist for non-sequential models. Non-sequential rays can be used to model the ordinary and extraordinary beams at the same time, and will account for any interference in the overlapping regions.

Modes 0, 1, 2, and 3 for the sequential "Birefringent In" surface are shown in the graphic below:
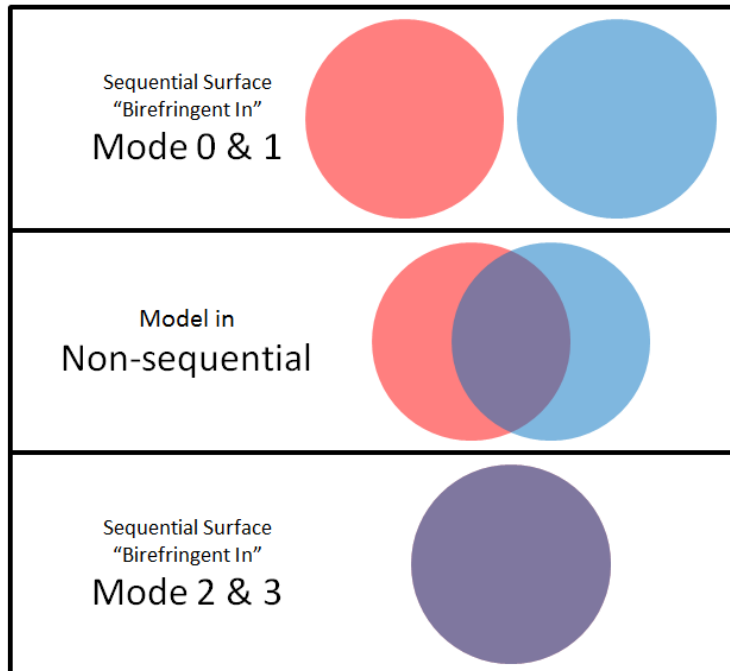
Transmission and other properties of birefringent surfaces

When light propagates through birefringent media, the index of refraction of the glass is different for the S and P polarizations. The ordinary index is what is seen by the perpendicular, or S-polarized light, while the effective index is seen by the parallel or P-polarized light. Note the S and P polarization directions used in this context are not in general the same as those used by the coating and Fresnel surface effects computation. Here S and P refer to the perpendicular and parallel orientations relative to the crystal axis rather than the surface normal vector. The plane that contains the refracted ray and the crystal axis vector is the parallel plane; and the P vector lies in this plane normal to the ray vector. The S vector is perpendicular to both P and the ray vector.

For polarization analysis of birefringent materials, a few assumptions are needed. If the mode is 0, the ordinary ray is traced, which only has an S component, so the P component transmission is set to zero. If the mode is 1, the extraordinary ray is traced, and the S component is therefore set to zero. This technique yields the correct transmission results for each possible path separately. However, to get the total transmission requires analysis of each possible combination of modes for every pair of birefringent surfaces. If there are 2 pairs of birefringent surfaces in the system, 4 separate ray traces are required; and if there are 3 pairs of birefringent surfaces, 8 traces required, etc.

If the mode is 2, then the ray follows the ordinary path, but the ordinary index is used to phase rotate the S component of the field, and the effective index (properly referenced to the ordinary ray direction) is used to rotate the P component of the electric field. Only 1 trace is needed to properly model both transmission and phase rotation. If the mode is 3, then the ray follows the extraordinary path, with a similar phase rotation as in the case of mode 2.

When modeling bulk transmission through birefringent media, only the internal transmittance as defined by the ordinary index data is considered. No accounting for dichroism is supported.

Sample Zemax files

For some samples of the birefringent surface type, see the "<data>\Samples\Sequential\Birefringent prisms" folder. Shown below is the "Wollaston prism" sample file showing two configurations superimposed. The two halves of the prism are both birefringent, with the crystal axis oriented at 90 degrees to each other.
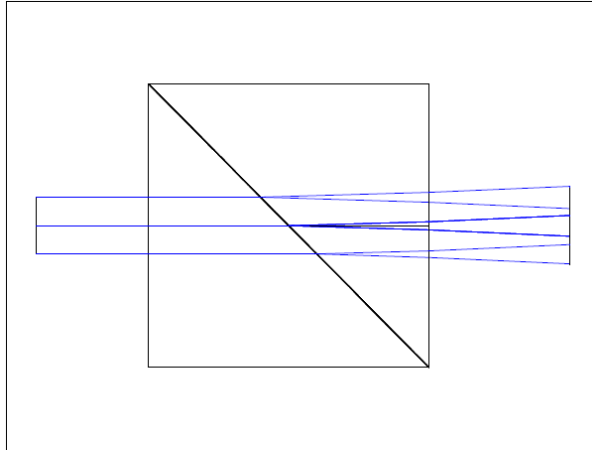
PARAMETER DEFINITIONS FOR BIREFRINGENT "IN" SURFACES

Parameter 0	Parameter 1	Parameters 2,3,4	Parameter 5	Parameter 6
Length to draw crystal axis, 0 to not draw.	Mode, 0, 1, 2, or 3.	X, Y, and Z-cosines	Parax Ignore. If 1, paraxial rays ignore the extraordinary index.	Shape. If the shape is not zero, parameters 7 and up are used to define the surface shape

PARAMETER DEFINITIONS FOR BIREFRINGENT "OUT" SURFACES

Parameter 0-5	Parameter 6
Unused.	Shape. If the shape is not zero, parameters 7 and up are used to define the surface shape

Black Box Lens

The Black Box Lens surface is used to "play back" a Zemax Black Box (BB) file previously created using the export Black Box data tool, described in "Export Zemax Black Box Data".

To load a BB file, set the surface type to "Black Box Lens" and then enter the name of the file, with the extension ZBB, in the comment column of the Lens Data Editor. The file must be placed in the <data>\BlackBoxes folder. Zemax will load the file, decrypt the prescription data contained within the file, and then use the data to trace rays through the hidden prescription data whenever a ray is incident upon that surface.

If the first surface of a lens file is a BB surface, and the object is at infinite conjugates, insert a dummy STANDARD surface in front of the BB surface. This is done to simplify how Zemax internally handles systems at infinite conjugates. The surface before and after the BB may have any catalog glass defined; model and table glasses are not supported. The BB surface itself cannot have a glass type. The thickness, aperture type, and aperture size of the BB are automatically set by the data defined within the ZBB file, and cannot be modified by the user. The object and image surfaces may not be BB surfaces.

Features not supported when using Black Box surfaces

When using BB surfaces, not all analysis features are supported to such an extent as to be able to reproduce the analysis data for the original system. These include features where the individual surface data must be modified to complete the analysis, such as for the thermal analysis features. Also, analysis features that require significant surface-by-surface computations cannot yield the same analysis data as the original file, and are therefore not supported. These include Physical Optics Propagation and Skew Gaussian Beam analysis. It is highly recommended that the creator of the ZBB data verify that the computations of interest to the end user of the ZBB work as they did for the original file, and provide guidance to the end user as to what aspects of the lens performance the BB is intended to illustrate. All common features that rely directly on ray tracing, such as ray fans, OPD fans, spot diagrams, MTF, PSF, image analysis, and most other analysis features work well.

Conjugate

The conjugate surface is defined by two user-specified points. Zemax always uses the surface vertex as the reference point; the two points required to define the conjugate surface are specified relative to this vertex. The conjugate surface will always perfectly image one point to the other point, assuming the surface is a mirror. Although the conjugate surface can have any material type, it is useful to think of it as being defined by its reflective properties.

If the z-coordinates of the two points are either both positive or both negative, then the image formed from one of the points to the other is real. In this case, the distance from one of the points to an arbitrary point on the surface, plus the distance from the arbitrary point on the surface to the second point, is constant for all points on the surface. One additional constraint is needed to make the surface unique: the surface must pass through the vertex of the local coordinate system. If the surface is reflective, then one point is the conjugate of the other, hence the name.

The surface generated by these two points satisfies the following expression if both z1 and z2 have the same sign:

$$\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} + \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} = \sqrt{x_1^2+y_1^2+z_1^2} + \sqrt{x_2^2+y_2^2+z_2^2}.$$

Note that the surface must intersect the point (0,0,0). Several types of surfaces can be formed with this model. For example, a sphere can be formed by setting the x and y values to zero, and the two z values each to the radius of the sphere. An elliptical surface of arbitrary orientation can be formed by specifying non-zero values for either the x or y values.

If z1 and z2 have opposite signs, then the image formed from one of the points to the other is virtual. In this case, the distance from one of the points to an arbitrary point on the surface, minus the distance from the arbitrary point on the surface to the second point, is constant for all points on the surface. Like the real imaging case, the surface must pass through the vertex of the local coordinate system.

The surface generated by these two points satisfies the following expression if z1 and z2 have opposite signs:

$$\sqrt{(x-x_1)^2+(y-y_1)^2+(z-z_1)^2} - \sqrt{(x-x_2)^2+(y-y_2)^2+(z-z_2)^2} = \sqrt{x_1^2+y_1^2+z_1^2} - \sqrt{x_2^2+y_2^2+z_2^2}.$$

Note that the surface must intersect the point (0,0,0). Several types of surfaces can be formed with this model. For example, a hyperbola can be formed by setting the x and y values to zero, and the two z values to opposite values. If the z values are equal but opposite, then a plane will be generated.

The coordinates of the two construction points are specified in the parameter columns, as shown in the following table. Neither the z1 nor the z2 value can be zero.

PARAMETER DEFINITIONS FOR CONJUGATE SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
x_1	y_1	z_1	x_2	y_2	z_2

Coordinate Break

The coordinate break surface is used to define a new coordinate system in terms of the current system. It is always considered a "dummy" surface for ray tracing purposes. There are six parameters used to describe the new coordinate system: x-decenter, y-decenter, tilt about x, tilt about y, tilt about z, and a flag to indicate the order of tilting and decentration. Coordinate break decenters are always specified in lens units. Coordinate tilts are specified in degrees, and are right-handed with respect to the positive axes. Coordinate breaks are always relative to the coordinate system of the previous surface.

An alternate way of implementing coordinate breaks is to use surface tilts and decenters, see "Surface tilt/ decenter tab" for details.

The order flag

> *The order of the decenters and tilts matters!*

If the "order" flag is set to zero, Zemax first decenters in x and y (decenters are orthogonal so the order does not matter). Then Zemax tilts about the local x axis (which rotates the y and z axes to new orientations), then tilts about the new y axis (which rotates the x and z axes), then finally tilts about the new z axis.

If the "order" flag is any other value (such as unity), then the tilts are done first, in the order around local z, then around the new y, and then around the new x, and then finally the decenters are done. This "order" flag is extremely useful because a single coordinate break can undo an earlier coordinate break, even for compound tilts and decenters.

The coordinate break acts like a plane surface oriented in the coordinate system after the decenters and tilts have been applied. However, the surface is never drawn, and cannot be used to define the boundary between two media. The glass type will always be the same as the prior surface, and Zemax will display "-" for the glass name, which is meant to indicate that a glass type cannot be entered there. Coordinate breaks themselves can never be mirrors, nor can the object surface be a coordinate break.

PARAMETER DEFINITIONS FOR COORDINATE BREAK SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
Decenter X	Decenter Y	Tilt About X	Tilt About Y	Tilt About Z	Order

Cubic Spline

The cubic spline surface is described by sag values which are the distances between the vertex tangent plane and the surface. Spline surfaces are used to describe unusual correctors, headlamps, and other non-standard optical surfaces, but rarely for imaging applications because of the fundamental properties of splines. See "Comments about spline surfaces" below for more discussion.

The Cubic Spline surface uses eight values to represent the sag at one-eighth, two-eighths, and so on to eight- eighths of the semi-diameter of that surface. Cubic spline surfaces are rotationally symmetric. All eight points must be defined. A subset cannot be used, although the semi-diameter may be defined to exceed the useful aperture of the surface. This is often required because of the steep curvatures occasionally introduced by spline fitting. If eight points provides an overly coarse sampling, see "Extended Cubic Spline". For a more general non-rotationally symmetric surface, see "Grid Sag".

Comments about spline surfaces

Cubic splines are formed by a piece-wise concatenation of curved segments. Within the bounds of each segment, the curve is defined by a third order polynomial. The polynomial coefficients describing each segment are determined from the sag values of the defined segment boundaries. The determination of the coefficients is driven by the boundary requirements that the curve goes through the defined points, and both the first and second derivatives be continuous across segment boundaries. For a third order spline, it is not possible to require higher order derivatives, such as third, to be continuous across segment boundaries. For this reason, splines are of limited accuracy and usefulness in high precision optical design.

A common characteristic of tracing rays through spline surfaces is rough or noisy looking ray data, with discontinuities in some results. These ray trace discontinuities are a fundamental limitation of splines, and they are not due to a flaw in Zemax, or a lack of numerical precision.

Higher order splines of course exist, and one way to eliminate the discontinuities is to use a higher order spline and fewer segments. In the limit, this is essentially the same as using a single high order polynomial for the whole surface, see for example the "Even Asphere". This is why high order polynomials, and not splines or NURBS, dominate in precision optical design; they are continuously smooth and differentiable to all orders.

For an excellent discussion of spline theory, properties and algorithms, see Numerical Recipes in C, by Press et al., Cambridge University Press.

PARAMETER DEFINITIONS FOR CUBIC SPLINE SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Sag at 1/8	Sag at 2/8	Sag at 3/8	Sag at 4/8	Sag at 5/8	Sag at 6/8	Sag at 7/8	Sag at 8/8

Cylinder Fresnel

The Cylinder Fresnel surface has a polynomial aspheric cylindrical substrate with an independent polynomial aspheric cylindrical function defining the Fresnel lens properties. The surface substrate sag is given by:

$$z_s = \frac{c_s y^2}{1 + \sqrt{1 - (1 + k_s)c_s^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14} + \alpha_8 y^{16}.$$

The previous expression is used to compute the ray-surface intercept. Once the intercept is found, the refraction (or reflection) of the surface is determined by the local slope of the Fresnel facets. The Fresnel facet shape is described by an identical expression, with independent coefficients:

$$z_f = \frac{c_f y^2}{1 + \sqrt{1 - (1 + k_f)c_f^2 y^2}} + \beta_1 y^2 + \beta_2 y^4 + \beta_3 y^6 + \beta_4 y^8 + \beta_5 y^{10} + \beta_6 y^{12} + \beta_7 y^{14} + \beta_8 y^{16}.$$

Note that the curvature (symbol c), the conic constant (symbol k), and all of the polynomial coefficients are independent for the substrate sag and Fresnel portions of the surface. The refraction at the surface accounts for only the Fresnel sag, while the ray-surface intercept depends only upon the substrate sag.

The substrate sag radius, conic, and polynomial terms are all specified in the Lens Data Editor. The Fresnel sag terms are specified as extra data values in an extra data file (see the Import section of the Surface Properties). However, the extra data file uses curvature (the reciprocal of radius) rather than radius for the Fresnel sag. The Parameter and Extra Data Values used are summarized in the following tables.

PARAMETER DEFINITIONS FOR CYLINDER FRESNEL SURFACES

Parameters 1-8
$\alpha_1 - \alpha_8$

EXTRA DATA DEFINITIONS FOR CYLINDER FRESNEL SURFACES

Parameter Number	Description
9-12	Unused
13	Maximum aspheric polynomial term number, n. The maximum is 8
14	The Y direction curvature (NOT RADIUS) of the Fresnel surface. This parameter will affect the refraction of the surface, but not the shape of the substrate.
15	The Y direction conic constant of the Fresnel surface. This parameter will affect the refraction of the surface, but not the shape of the substrate.
16	Coefficient on y^2
n	Coefficient on $y^{2(n-15)}$

The "Maximum aspheric polynomial term number" is used to specify the maximum polynomial term to be used in calculating the Fresnel sag. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored. As with any complex surface model, extreme care should be taken to evaluate the accuracy and appropriateness of this model, especially where fabrication decisions are concerned.

Because there is no reliable way to compute the phase through a Fresnel surface which is not a plane, any calculation that requires OPD data, such as OPD fans, MTF, and Zernike coefficients, will not be supported if a non-plane substrate Fresnel surface is present in the lens description.

Data

The data surface is a dummy surface that provides up to 240 extra data values to a subsequently defined user- defined surface (UDS). The data surface must be placed immediately before the UDS and up to 20 such surfaces defined sequentially may be used by a single UDS. The purpose of this surface is to allow a large set of user- definable values to be parametrically controlled and optimized, such as grids of data points. This capability requires the FIXED_DATA4 UDS DLL structure.

EXTRA DATA DEFINITIONS FOR DATA SURFACES

Parameter Number	Description
13 - 252	Data values.

Diffraction Grating

The diffraction grating surface can be used to model straight-line ruled gratings. The grating lines are parallel to the local x-axis. Other orientations can be simulated by using a coordinate break surface before and after the grating surface. For a plane grating, rays traced to the grating are refracted according to the equation

$$n_2 \sin\theta_2 - n_1 \sin\theta_1 = \frac{M\lambda}{d} = M\lambda T,$$

where d is the grating spacing (always in micrometers), θ_2 is the refracted angle, θ_1 is the incident angle, M is the diffraction order, λ is the wavelength (always in micrometers), n_1 and n_2 are the indices of refraction before and after the grating, and T is the grating frequency in lines per micrometer. Note that the sign convention for M is completely arbitrary. Zemax uses the definition for T (lines per micrometer) rather than d (micrometers per line). The grating surface can be plane, spherical, or conical, and the medium before the grating, as well as the grating itself, can be air, glass, "MIRROR" or any other valid glass type. The grating is described by the y-spacing of the grating lines measured in lines per micrometer (independent of the system units) and the diffraction order. Zemax only models gratings to the extent of deviating ray paths. Other properties, such as efficiency, and relative transmission are not supported. If the grating spacing is too small (or if T is too large) to satisfy the grating relation, then a "Ray missed surface" error will be reported.

PARAMETER DEFINITIONS FOR DIFFRACTION GRATING SURFACES

Parameter 1	Parameter 2
Grating lines per micrometer	Diffraction order

Elliptical Grating 1

The elliptical grating 1 surface is a combination of an elliptical surface with polynomial aspheric terms and a grating with grooves that are straight when projected on to the vertex tangent plane (although the surface itself is usually not planar). The grating lines are parallel to the local x-axis, and may be equally spaced or have a variable spacing defined by a 4 term polynomial (up to the cubic term in y). Other orientations can be simulated by using a coordinate break surface before and after the grating surface. For a plane grating, rays traced to the grating are refracted according to the equation

$$n_2 \sin\theta_2 - n_1 \sin\theta_1 = \frac{M\lambda}{d} = M\lambda T,$$

where d is the effective grating spacing (always in micrometers), θ_2 is the refracted angle, θ_1 is the incident angle, M is the diffraction order, λ is the wavelength (always in micrometers), n_1 and n_2 are the indices of refraction before and after the grating, and T is the effective grating frequency in lines per micrometer. Zemax uses the definition for T (lines per micrometer) rather than d (micrometers per line). The effective grating frequency T is defined by:

$$\frac{1}{T} = d_{eff} = \frac{1}{T_0} + \alpha y + \beta y^2 + \Gamma y^3 + \Delta y^4 + \varepsilon y^5.$$

Note the constant spacing term is defined in reciprocal units to be compatible with other grating surfaces in Zemax; while the higher order terms are defined in a polynomial in the y coordinate. If T evaluates to zero, the grating component is ignored, which is useful for modeling elliptical aspheres which are not gratings.

The medium behind the grating surface may be air, glass, or a mirror. Zemax only models gratings to the extent of deviating ray paths. Other properties, such as efficiency and relative transmission are not supported. If the grating spacing is too small (or if T is too large) to satisfy the grating relation, then a "Ray missed surface" error will be reported.

Elliptical Grating surface shape

The surface shape is an ellipse with added polynomial aspheric terms. The sag of the surface is given by the following expression:

$$z = \frac{cu^2}{1 + \sqrt{1 - u^2}} + \sum_{i=1}^{N} A_i E_i(x, y),$$

where $u^2 = a^2 x^2 + b^2 y^2$.

In these expressions, x, y, and z are the ray-surface intercept coordinates, a, b, and c are the coefficients which define the shape of the ellipse, and the extended polynomials represented by E are defined in the section "Extended Polynomial".

Note the symbols a, b, and c do not have the same meaning in this sag expression as these same symbol names do in the description of the Standard surface. To make a spherical elliptical surface, the values a and b should be identical and equal to one over the radius of curvature of the surface, and the value c should be equal to the radius of curvature:

$$a = b = \frac{1}{R} = \frac{1}{c}.$$

PARAMETER DEFINITIONS FOR ELLIPTICAL GRATING 1 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameters 6-10
Lines per micrometer, T_0	Diffraction order	a	b	c	$\alpha, \beta, \Gamma, \Delta, \varepsilon$

EXTRA DATA DEFINITIONS FOR ELLIPTICAL GRATING 1 SURFACES

Parameter Number	Description
11-12	Unused
13	Maximum term number.
14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation
15 – 244	Polynomial terms
245 and above	Unused

Elliptical Grating 2

The elliptical grating 2 surface is very similar to the elliptical grating 1 surface. The distinction is that the grating lines are formed by the intersection of tilted, parallel, evenly spaced planes with a substrate defined by an elliptical surface with polynomial aspheric terms. The intersection of the surface and tilted planes yield a grating with grooves that are straight when projected on to a plane tilted by an angle θ_y, measured in the local y direction. When projected onto a surface which is not plane, the grooves are curved, not straight.

The grating is defined by the parameter T, which is the lines per micrometer (the grating frequency) when the groove lines are projected onto the tilted XY plane. This is the reciprocal of the spacing between the planes which form the grooves. The diffraction order is M, and the elliptical substrate is defined by the parameters a, b, c, and extra data terms, exactly as for the elliptical grating.

The medium behind the grating surface may be air, glass, or a mirror. Zemax only models gratings to the extent of deviating ray paths. Other properties, such as efficiency and relative transmission are not supported. If the grating spacing is too small (or if T is too large) to satisfy the grating relation, then a "Ray missed surface" error will be reported. The surface shape is an ellipse with added polynomial aspheric terms, see the "Elliptical Grating surface shape".

PARAMETER DEFINITIONS FOR ELLIPTICAL GRATING 2 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
Lines per micrometer, T	Diffraction order	a	b	c	θ_y

EXTRA DATA DEFINITIONS FOR ELLIPTICAL GRATING 2 SURFACES

Parameter Number	Description
7-12	Unused
13	Maximum term number.
14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation.
15 – 244	Polynomial terms
245 and above	Unused

Even Asphere

Rotationally symmetric polynomial aspheric surfaces are described by a polynomial expansion of the deviation from a spherical (or aspheric described by a conic) surface. The even asphere surface model uses only the even powers of the radial coordinate to describe the asphericity. The model uses the base radius of curvature and the conic constant. The surface sag is given by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}.$$

Note that the coefficients have units. The coefficients are entered in the corresponding parameter columns, as shown in the following table.

PARAMETER DEFINITIONS FOR EVEN ASPHERE SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8

Extended Asphere

The extended asphere surface is similar to the Even Asphere surface; see "Even Asphere". However, the extended asphere can support aspheric coefficients up to order 480, whereas the even asphere is limited to 16. Also, a slightly different method is used for computing the coefficients of the polynomial terms. The sag of the surface is given by the following expression:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \sum_{i=1}^{N} \alpha_i \rho^{2i},$$

where the first expression is identical to the Standard surface, and the second term is a power series sum over a normalized radial coordinate. The normalized radial coordinate ρ is used because then the coefficients α_i all have units of lens units. The extra data terms are defined in the following table.

EXTRA DATA DEFINITIONS FOR EXTENDED ASPHERE SURFACES

Parameter Number	Description
13	Maximum term number. The maximum is 240.
14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation
15	Coefficient in lens units on ρ^2
16	Coefficient in lens units on ρ^4
n	Coefficient in lens units on $\rho^{2(n-14)}$

The "Maximum term number" is used to specify the maximum polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored. For example, if the last term in the series you wish to use is ρ_{14} , which is term number 7, then specify "7" in the maximum term number column. Note that the term number is 7, because it is the 7th term in the polynomial expansion.
The normalization radius scales the X and Y intercept coordinates of the ray so that the polynomial terms are all dimensionless and the coefficients are all in lens units. For modeling surfaces that are not well described by polynomials, see "Grid Sag" or "User Defined".

Extended Cubic Spline

The Extended Cubic Spline surface is similar to the Cubic Spline surface (see "Cubic Spline" 2), with the key difference being more terms are allowed. The Extended Cubic spline supports between 4 and 240 terms. The surface sag is defined by a list of sag points evenly spaced along the radius of a rotationally symmetric shape. The sag must always be zero at zero radius, and so this point is not entered into the spreadsheet. For important information about the limitations of splines, see "Comments about spline surfaces" . For a non-rotationally symmetric spline surface, see "Grid Sag".

If you have a list of sag points that describes your optical surface (as might be generated from a computer program other than Zemax) and you wish to enter them into Zemax, use the Import section of the Surface Properties dialog. This feature is described in "Extra Data".

EXTRA DATA DEFINITIONS FOR EXT. CUBIC SPLINE SURFACES

Parameter Number	Description
13	Maximum number of spline points.
14	Step size in lens units between points along the radius.
15	The sag at a distance of one step size away from the vertex
16	The sag at a distance of 2 step sizes away from the vertex.
n	The sag at a distance of (n-14) step sizes away from the vertex

The "Maximum number of spline points" is used to specify how many points define the surface. This number may be between 4 and 240. At least 4 points are required to define the curved surface.

Spline surfaces can cause rough or ragged ray tracing results. A more general and much smoother solution is to use the grid sag surface, which is not restricted to rotational symmetry. See "Grid Sag".

Extended Fresnel

The Fresnel surface described in "Fresnel" models a plane surface which has the refractive (or reflective) power of a curved surface. The Extended Fresnel surface increases the flexibility of this model by supporting a plane, spherical, conic, or polynomial aspheric substrate on which a plane, spherical, conic, or polynomial Fresnel pattern is etched.

The surface sag is identical to the even aspheric surface:

$$ z_s = \frac{c_s r^2}{1 + \sqrt{1 - (1 + k_s)c_s^2 r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}. $$

See the section "Even Asphere" for details. The previous expression is used to compute the ray- surface intercept. Once the intercept is found, the refraction (or reflection) of the surface is determined by the local slope of the Fresnel facets, which depends upon both the Fresnel facet shape expression for Zf (below) and the substrate shape expression for Zs (above). The Fresnel facet shape is described by an expression virtually identical to the even asphere sag expression:

$$ z_F = \frac{c_F r^2}{1 + \sqrt{1 - (1 + k_F)c_F^2 r^2}} + \sum_{i=0}^{n} \alpha_i r^{2i}. $$

The only difference is that the latter expression need not use all 8 terms if n is less than 8, the maximum accepted value. Note that the curvature (symbol c), the conic constant (symbol k), and all of the polynomial coefficients (symbol α) are independent for the substrate sag and Fresnel portions of the surface. The refraction at the surface accounts for both the

substrate sag and the Fresnel sag, while the ray-surface intercept depends only upon the substrate sag. The intention was to model a Fresnel lens molded on a plane which is curved or warped into a new substrate shape after it is fabricated.

The substrate sag radius, conic, and polynomial terms are all specified in the Lens Data Editor, just like the even asphere surface. The Fresnel sag terms are specified as extra data values in an extra data file (see the Import section of the Surface Properties). However, the extra data file uses curvature (the reciprocal of radius) rather than radius for the Fresnel sag. The Extra Data Values used are summarized in the following table.

EXTRA DATA DEFINITIONS FOR EXTENDED FRESNEL SURFACES

Parameter Number	Description
13	Maximum aspheric polynomial term number, n. The maximum is 8.
14	The curvature (NOT RADIUS) of the Fresnel surface. This parameter will affect the refraction of the surface, but not the shape of the substrate
15	The conic constant of the Fresnel surface. This parameter will affect the refraction of the surface, but not the shape of the substrate.
16	Coefficient on r^2
n	Coefficient on $r^{2(n-15)}$

The "Maximum aspheric polynomial term number" is used to specify the maximum polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored. As with any complex surface model, extreme care should be taken to evaluate the accuracy and appropriateness of this model, especially where fabrication decisions are concerned.

Because there is no reliable way to compute the phase through a Fresnel surface which is not a plane, any calculation that requires OPD data, such as OPD fans, MTF, and Zernike coefficients, will not be supported if a non-plane substrate Fresnel surface is present in the lens description.

Extended Odd Asphere

The Extended Odd Asphere surface is similar to the Odd Asphere surface described in "Odd Asphere". However, the Extended Odd Asphere can support aspheric coefficients up to order 240, whereas the Odd Asphere is limited to 8. Also, a slightly different method is used for computing the coefficients of the polynomial terms. The sag of the surface is given by the following expression:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=1}^{N} \alpha_i \rho^i,$$

where the first expression is identical to the Standard surface, and the second term is a power series sum over a normalized radial coordinate. The normalized radial coordinate ρ is used because then the coefficients α_i all have units of lens units. The extra data terms are defined in the following table.

EXTRA DATA DEFINITIONS FOR EXTENDED ODD ASPHERE SURFACES

Parameter Number	Description

13	Maximum term number. The maximum is 240.
14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation
15	Coefficient in lens units on ρ^1
16	Coefficient in lens units on ρ^2
n	Coefficient in lens units on ρ^{n-14}

The "Maximum term number" is used to specify the maximum term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored.

Extended Polynomial

The extended polynomial surface is similar to the polynomial surface, except more terms are allowed. The surface also supports a base conic asphere surface upon which the polynomial aspheric terms are added. The surface sag is of the form:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=1}^{N} A_i E_i(x, y).$$

where N is the number of polynomial coefficients in the series, and A_i is the coefficient on the ith extended polynomial term. The polynomials are a power series in x and y. The first term is x, then y, then x*x, x*y, y*y, etc. There are 2 terms of order 1, 3 terms of order 2, 4 terms of order 3, etc. The maximum order is 20, which makes a maximum of 230 polynomial aspheric coefficients. The position values x and y are divided by a normalization radius so the polynomial coefficients are dimensionless.

For example, the 12th term in the polynomial expansion, which is extra data term number 14, is the coefficient on the term in $x^2 y^2$. The coefficients A_i all have units which are the same as the lens units, such as millimeters or inches.

EXTRA DATA DEFINITIONS FOR EXTENDED POLYNOMIAL SURFACES

Parameter Number	Description
13	Maximum term number.
14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation
15 -244	Polynomial terms.
245 and above	Unused

The "Maximum term number" is used to specify the maximum polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored. For example, if the last term in the series you wish to use is xy^3, which is term number 13, then specify "13" in the maximum term number column. Note that the term number is 13, because it is the 13th term in the polynomial expansion. The normalization radius scales the X and Y intercept coordinates of the ray so that the polynomial terms are all dimensionless and the coefficients are all in lens units.

Extended Toroidal Grating

Extended toroidal grating surfaces are a combination of the toroidal, toroidal grating, and extended polynomial surfaces. Toroidal surfaces are described by defining a curve in the Y-Z plane, and then rotating this curve about an axis parallel to the Y axis and intersecting the Z axis. The resulting shape is then deformed by the addition of extended polynomials. Finally, a linear diffraction grating is placed upon the surface. The curve in the Y-Z plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14} + \alpha_8 y^{16}.$$

This curve is similar to the even asphere surface sag formula, except the coordinate argument is y, not r . This curve is then rotated about an axis a distance R from the vertex. This distance R is referred to as the radius of rotation, and may be positive or negative. The Y-Z radius of curvature is specified in the same column on the spreadsheet editor as the radius for Standard surfaces. The radius of rotation is set on parameter column number 1. To model a cylinder lens which is flat in the X direction use zero, which Zemax interprets as infinite radius.

The resulting surface is then modified by the addition of the extended polynomial terms:

$$z = z_{toroid} + \sum_{i=1}^{N} A_i E_i(x, y).$$

where N is the number of polynomial coefficients in the series, and A_i is the coefficient on the ith extended polynomial term. The polynomials are a power series in x and y. The first term is x, then y, then x*x, x*y, y*y, etc. There are 2 terms of order 1, 3 terms of order 2, 4 terms of order 3, etc. The maximum order is 20, which makes a maximum of 230 polynomial aspheric coefficients. The position values x and y are divided by a normalization radius so the polynomial coefficients are dimensionless.

For example, the 12th term in the polynomial expansion, which is extra data term number 14, is the coefficient on the term in $x^2 y^2$. The coefficients A_i all have units which are the same as the lens units, such as millimeters or inches.

The diffraction grating is defined in terms of the number of lines per micrometer and the diffraction order. These values are specified in parameter columns 2 and 3, respectively. The grating lines are parallel to the local x axis, and are evenly spaced when projected onto a plane.

PARAMETER DEFINITIONS FOR EXTENDED TOROIDAL GRATING SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameters 4-11
Radius of Rotation	Grating Lines per micrometer	Diffraction order	$\alpha_1 - \alpha_8$

EXTRA DATA DEFINITIONS FOR EXTENDED TOROIDAL GRATING SURFACES

Parameter Number	Description
12	Unused
13	Maximum term number.
14	Normalization radius. All ray-intercept points are divided by this

	number to determine the x and y coordinates for polynomial evaluation
15 -244	Polynomial terms.
245 and above	Unused

The "Maximum term number" is used to specify the maximum polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored. For example, if the last term in the series you wish to use is xy^3, which is term number 13, then specify "13" in the maximum term number column. Note that the term number is 13, because it is the 13th term in the polynomial expansion. The normalization radius scales the X and Y intercept coordinates of the ray so that the polynomial terms are all dimensionless and the coefficients are all in lens units.

Filter

Filter surfaces are modeled using the User Defined Surface described in "User defined surface apodization using DLLs".

Fresnel

The Fresnel surface model is used to simulate flat surfaces which have been etched to have a spherical (or optionally aspheric) profile on a small scale (for Fresnel lenses which are not flat see the extended Fresnel surface type description). The surface intercept is determined by computing the intersection of the incoming ray with a plane. Once the plane intercept points are found, the surface is then treated as generally curved for the purposes of refraction into the next medium. This is only an approximation to a real Fresnel lens, however. The real Fresnel lens has grooves which may alter the exact intercept point. The model used here is adequate for Fresnel lens which have fine grooves (the groove depth is very shallow compared to the aperture). Extreme Fresnel lenses, such as those used in lighthouses, are not modeled well. The radius of curvature and conic constant, if any, are specified in the same manner as a Standard surface. The parameter values are exactly the same as for the even asphere model; polynomial aspherics are supported to 16th order.

For other types of Fresnel surfaces, see "Extended Fresnel" and "Generalized Fresnel".

PARAMETER DEFINITIONS FOR FRESNEL SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8

Generalized Fresnel

This surface represents a generalization of the Fresnel surface described in "Fresnel". The basic concept of a Fresnel surface, as far as ray tracing is concerned, is that the overall shape of the surface substrate is independent of the local slope. The generalized Fresnel uses a polynomial aspheric substrate model, identical to the Even Aspheric surface described in "Even Asphere". The sag of the surface is given by the following expression:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16},$$

where the terms are identical to those in the even asphere surface model. See that section for a complete discussion.

Once the ray has intercepted the surface, the ray reflects or refracts as if the surface had a shape described by a polynomial expansion of the form:

$$z = \sum_{i=1}^{N} A_i E_i(x, y),$$

where N is the number of polynomial coefficients in the series, and A_i is the coefficient on the ith extended polynomial term. The polynomial terms are identical to those described in "Extended Polynomial". It is the gradient of the extended polynomial terms alone that determines the reflective or refractive properties of the surface, not the surface substrate shape.

One application for this surface type is modeling faceted surfaces. For example, a flat substrate may consist of a series of small faceted planes, which would reflect or refract the light as though the surface was tilted. This can be simulated using a flat substrate and a linear x or y tilt term in the polynomial coefficients.

Because there is no reliable way to compute the phase through a Fresnel surface which is not a plane, any calculation that requires OPD data, such as OPD fans, MTF, and Zernike coefficients, will not be supported if a non-plane substrate Fresnel surface is present in the lens description.

Note that the coefficients have units. The coefficients are entered in the corresponding parameter columns, as shown in the following table.

PARAMETER DEFINITIONS FOR GENERALIZED FRESNEL SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
α_1	α_2	α_3	α_4	α_5	α_6	α_7	α_8

EXTRA DATA DEFINITIONS FOR GENERALIZED FRESNEL SURFACES

Parameter Number	Description
9-12	Unused
13	Maximum term number.
14	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation
15 -244	Polynomial terms.
245 and above	Unused

The "Maximum term number" is used to specify the maximum polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation, as terms beyond this number are ignored.

Gradient 1

This surface has the same shape as the Standard surface, with media whose index of refraction is described by

$$n = n_0 + n_{r2}r^2 + n_{r1}r,$$

where $r^2 = x^2 + y^2$. Four parameters are required: the maximum step size, Δt, the base index, n_0, the radial quadratic index, n_{r2}, and the radial linear index, n_{r1}. Note that n_{r2} and n_{r1} have units.

Discussion on maximum step size for GRIN surfaces

The maximum step size is measured in lens units along the local z-axis and determines the increments in which the gradient index is recalculated. For example, if $\Delta t = 0.1$ and the system units are millimeters, then the index of refraction will be recalculated every 0.1 mm along the z-axis. Therefore, a ray passing through this gradient surface will refract every 0.1 mm.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. The exact value required depends upon the numerical aperture of the system and the magnitude of the coefficients. To determine a suitable step size, start at a large value (on the order of the thickness of the surface), and then perform a spot diagram. Note the RMS spot radius. Now decrease the step size by a factor of two. If the RMS spot radius changes by less than a few percent, the new step size is probably small enough. Otherwise, decrease the step again. For the final design phase, you may want to decrease the step size again. Using too small a step size needlessly slows the ray tracing speed without enhancement of accuracy. OPD tracing generally converges more slowly than ray tracing, so you may want to repeat the above procedure while inspecting the OPD fan. Occasionally check to ensure that the step size is appropriate as the design progresses.

Restrictions on surfaces following GRIN surfaces

The ray trace through a gradient index media requires iteration to determine the point of intersection of the ray with the surface following the gradient index surface. Because of this, not all surface types are allowed to follow gradient index surface types. If the type of the surface after a gradient index surface is not supported, a message will be presented indicating the error. Support for additional surface types to follow a GRIN surface may be added upon request; contact Zemax Technical Support for details.

PARAMETER DEFINITIONS FOR GRADIENT 1 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4
Δt	n_0	n_{r2}	n_{r1}

Gradient 2

This surface has the same shape as the Standard surface, with media whose index of refraction is described by

$$n^2 = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{r6}r^6 + n_{r8}r^8 + n_{r10}r^{10} + n_{r12}r^{12},$$

where $r^2 = x^2 + y^2$. Eight parameters are required: the maximum step size, Δt, the base index squared, n_0, and the remaining six coefficients of the preceding equation. Note that some of the coefficients have units.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 2 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Δt	n_0	n_{r2}	n_{r4}	n_{r6}	n_{r8}	n_{r10}	n_{r12}

Gradient 3

This surface has the same shape as the Standard surface, with media whose index of refraction is described by

$$ n = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{r6}r^6 + n_{z1}z + n_{z2}z^2 + n_{z3}z^3 \; , $$

where $r^2 = x^2 + y^2$. Eight parameters are required: the maximum step size, Δt, the base index squared, n_0, and the remaining six coefficients of the preceding equation. Note that some of the coefficients have units.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 3 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Δt	n_0	n_{r2}	n_{r4}	n_{r6}	n_{z1}	n_{z2}	n_{z3}

Gradient 4

This surface has the same shape as the Standard surface, with media whose index of refraction is described by

$$ n = n_0 + n_{x1}x + n_{x2}x^2 + n_{y1}y + n_{y2}y^2 + n_{z1}z + n_{z2}z^2 \; . $$

Eight parameters are required: the maximum step size, Δt, the base index squared, n_0, and the remaining six coefficients of the preceding equation. Note that some of the coefficients have units. This particular GRIN model is useful for gradient index optics with a cylindrical power profile. It is also useful for modeling thermal gradients in optical elements, if you have sufficient data to compute the coefficients.

The linear transverse terms, n_{x1} and n_{y1}, are ignored when performing paraxial ray tracing.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 4 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Δt	n_0	n_{x1}	n_{x2}	n_{y1}	n_{y2}	n_{z1}	n_{z2}

Gradient 5

The shape of the Gradient 5 surface is the same as the Standard surface plus a "tilt" term in both x and y directions:

$$ z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + x\tan(\alpha) + y\tan(\beta) , $$

where c is the curvature (the reciprocal of the radius), r is the radial coordinate in lens units, k is the conic constant, and tan α and tan β are the tangents of the tilt angles in x and y. Note this is not the same surface shape as a tilted Standard

surface, but it is a close approximation as long as the curvature is small, or if the tilt angles are small. The Gradient 5 surface has the following gradient profile:

$$n_{ref} = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{z1}z + n_{z2}z^2 + n_{z3}z^3 + n_{z4}z^4,$$

where $r^2 = x^2+y^2$. Ten parameters are required: the maximum step size, Δt, the base index squared, n_0, and the remaining six coefficients of the preceding equation, and the two tilt terms. Note that some of the coefficients have units.

The important feature of the Gradient 5 model is that the model allows specification of dispersion properties of the media. The dispersion data is user defined and is stored in an ASCII file called SGRIN.DAT. The format for SGRIN.DAT will be described shortly.

The material name is entered in the glass column of the Gradient 5 surface type. If the glass column is blank, then dispersion effects are ignored.

To perform ray tracing, Zemax first computes the index at a "reference" wavelength using the preceding formula for n_{ref}. The index at any other wavelength is then computed using the following method based upon a general expansion of the Sellmeier formula:

$$n(\lambda)^2 = n(\lambda_{ref})^2 + \sum_{i=1}^{3} \frac{K_i(\lambda^2 - \lambda_{ref}^2)}{\lambda^2 - L_i}, \quad \text{where}$$

$$K_i = \sum_{j=1}^{K\_MAX} K_{ij}[n_{ref}]^{j-1}, L_i = \sum_{j=1}^{L\_MAX} L_{ij}[n_{ref}]^{j-1}, \text{ and } n_{ref} = n(\lambda_{ref}).$$

The coefficients K_{ij} and L_{ij} define the dispersion of the material, whereas the gradient coefficients specified by parameters 2-8 (see the following table) define the index gradient profile at the reference wavelength. This very general dispersion model permits modeling of nearly arbitrary gradient index dispersion over a wide wavelength band. The parameters K_MAX and L_MAX may be as little as 1 term, or up to 8 terms for more accurate modeling.

The dispersion data is stored in the ASCII file SGRIN.DAT which must reside in the <glass> folder (see "Folders").

The SGRIN.DAT file contains blocks of 10 lines each. The file format has the following structure:

```
MATERIALNAME
MIN_WAVELENGTH MAX_WAVELENGTH
REF_WAVELENGTH
K_MAX L_MAX
K11 K12 K13 ... K1K_MAX
K21 K22 K23 ... K2K_MAX
K31 K32 K33 ... K3K_MAX
L11 L12 L13 ... L1L_MAX
L21 L22 L23 ... L2L_MAX
L31 L32 L33 ... L3L_MAX
```

Multiple materials may be defined in the same file by placing additional blocks of 10 lines one after the other, with no blank lines between. The supplied file SGRIN.DAT contains coefficients to describe some gradient materials offered by LightPath Technologies.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 5 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8	Parameter 9	Parameter 10
Δt	n_0	n_{r2}	n_{r4}	n_{z1}	n_{z2}	n_{z3}	n_{z4}	$\tan\alpha$	$\tan\beta$

Gradient 6

This surface has the same shape as the Standard surface, with media whose index of refraction is described by

$$n = n_0 + n_1 r^2 + n_2 r^4 + n_3 r^6 + n_4 r^8.$$

The n_x values are computed from the following formula:

$$n_x = A_x + B_x \lambda^2 + \frac{C_x}{\lambda^2} + \frac{D_x}{\lambda^4},$$

with identical expressions (but different values for A, B, C, and D) for n_0, n_1, n_2, n_3 uses nanometers rather than micrometers for λ.

The dispersion data is user defined and is stored in an ASCII file called GLC.DAT. This file is stored in the folder <glass> (see "Folders"). The first line in the file is a file version format specifier of the form:

```
VERSION N
```

where N is the version number of the file format, currently 1. Zemax will read older GLC data files that omitted the version number, however the current version number and format should be used for any new files created or data added. The remaining lines are blocks of 21 lines defining the name and data for each material in the form:

```
NAME A0
B0
C0
D0
A1
B1
etc... ending with D4
```

The first line in the block is the name of the material, which can be any name (without special characters such as spaces or quotes) less than 10 characters long. The next 20 lines are the values for A, B, C, and D for n_0 through n_4, respectively. There are no blank lines or comments allowed. Zemax can read in data for up to 25 different materials in the GLC.DAT file.

The dispersion data in the supplied GLC.DAT file was provided by the Gradient Lens Corporation (GLC) in Rochester, NY, (716) 235-2620. Contact GLC for detailed information on material properties. Not all of the materials offered by GLC are included in the GLC.DAT file. The following materials are included: ARS10, ARS20, ARS27, and ARS31.

To use the Gradient 6 surface materials, change the surface type to Gradient 6, and then enter the appropriate material name in the glass column of the Lens Data Editor.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 6 SURFACES

Parameter 1
Δt

Gradient 7

This surface has the same shape as the Standard surface, with media whose index of refraction is described by

$$n = n_0 + \alpha(r - R) + \beta(r - R)^2,$$

where
$$r = \frac{R}{|R|}\sqrt{x^2 + y^2 + (R - z)^2}.$$

The coordinates x, y, and z are the usual coordinates measured with respect to the vertex tangent plane, and R is the radius of the isoindex contours measured at the vertex. The isoindex contours are spherical shells centered about the point z = R. The starting index, n_0, is measured at the vertex of the surface, not at the center of the isoindex contours. There are five parameters required: the maximum step size, Δt, the base index, n_0, R, α, and β. Note that α and β have units.

The isoindex contour radius R may be specified independently from the front or rear radius of curvature of the lens. However, if R is set to be zero, then Zemax assumes the isoindex radius and the front radius of curvature are equal.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 7 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
Δt	n_o	R	α	β

GRADIUM^{TM}

This surface type models lenses made from stock gradient index blanks available from LightPath Technologies, Inc. The surface shape is the same as the Standard surface. The blanks have an axial gradient index profile, which describes a reference index of refraction as a function of axial position within the glass blank. All that is required to define the lens is the starting position within the blank, the name of the stock blank profile, and of course the radii and thickness. The GRADIUM surface has a gradient index profile which is described by a polynomial of the form:

$$n = \sum_{i=0}^{11} n_i \left(\frac{z + \Delta z}{z_{max}}\right)^i.$$

The z coordinate is the distance from the front vertex of the surface, z_{max} is the maximum z coordinate of the blank (also known as the boule thickness), and the value Δz is the "offset" distance along the profile. Unlike most other gradient glass models in Zemax, the GRADIUM surface uses only fixed, predefined axial profile coefficients. The only design parameter required is the offset value, Δz. The available profiles provided with Zemax are defined in the ASCII file PROFILE.GRD. Other profile files may be defined, if they end in the extension GRD and are placed in the <glass> folder (see "Folders"). To select a different profile file, see "GRADIUM Profile". For a list of the available profiles, see "Gradium™ Profile".

GRADIUM profile file format

The file format is a series of blocks of 13 lines of data defined as follows:

```
PROFILE_NAME GLASS_FAMILY MAX_Z DENSITY UNUSED
n0
n1
...
n11
```

Each block of data starts with the name of the profile, which can be any valid ASCII name less than 20 characters. On the same line, the glass family name follows, which must be a gradient index material as defined in the SGRIN.DAT file, which is described in the section on the Gradient 5 surface type. The glass family name defines the reference wavelength which the profile describes. The last entry on the line is the maximum z coordinate of the blank. The 12 polynomial coefficients follow, from n0 to n11. The maximum allowed number of profiles is 100.

When Zemax performs ray tracing, the local z coordinate on the surface (which may be negative) is computed. The offset value is then added to determine where in the profile the coordinate lies.

Normally, the resulting value must always be positive and less than or equal to the maximum z, otherwise an error is generated (See the discussion of "Capping" below). The reference index is then evaluated. Once the reference index is computed, the index at the wavelength being traced is computed using the technique described in the section on the Gradient 5 surface.

The reference wavelength index at the front vertex is displayed as parameter 3. This value may be changed in the Lens Data Editor. When a new value is entered, Zemax computes the appropriate Δz to yield the specified reference index.

However, Δz is the value that matters. The reference index is just displayed for convenience, and should not be made variable, or a multi-configuration operand. Note the reference index is the vertex index at the reference wavelength, which is the wavelength defined in the glass family definition file SGRIN.DAT. It is not necessarily the primary wavelength.

The GRADIUM surface model also supports 4 additional parameter terms that are intended for use in tolerancing: Decenter X, Decenter Y, Tilt X, and Tilt Y. These four terms model an axial gradient that is not perfectly centered and not perfectly parallel to the local Z axis. The tolerance terms modify the axial profile by redefining the axial coordinate z as follows:

$$z' = t_x(x - d_x) + t_y(y - d_y) + t_z z,$$

$$\text{where} \quad t_z = [1.0 - t_x^2 - t_y^2]^{\frac{1}{2}},$$

and tx, ty, tz are the coefficients of the unit vector which points along the axial gradient axis and dx and dy are the decenters in lens units of the start of the profile. If tx and ty are both zero, then the dx and dy values do not matter (since the gradient is only along the z axis) and the tz value is unity. The terms tx and ty determine the slope of the profile axis in x and y, which is intended to model the tolerance of axial alignment between the gradient axis and the mechanical axis of the lens. This expression is a linear approximation which is only valid for asymptotically small tx and ty.

The tolerance terms, dx , dy , tx , and ty , are ignored when performing paraxial ray tracing.

Usually, only the defined range of the profile is used. However, in some cases the profile may be extended in one or both directions to add additional glass to the ends of the profile, which allows the use of GRADIUM in thicker lenses. This technique is called "capping". By default, Zemax turns capping off, so that any ray trace which requires glass beyond the profile limits is flagged as an error. This automatically enforces boundary constraints during optimization. To remove this restriction, the capping flag can be set to be 1, 2, or 3. The default value of zero indicates the blank is bounded to the profile length at both ends. If the capping flag is 1, then only the left edge is bounded (the right edge is allowed to pass beyond the profile limit). If the capping flag is 2, then only the right edge is bounded. And if the capping is 3, then neither the left nor right edges of the blank are bounded, and both the thickness and the offset may take on any values. The additional material added to the beginning and end of the profile is assumed to be a homogeneous glass with the same index and dispersion as the respective end points of the profile. This assumption may be inaccurate if there is some slope to the profile at the defined endpoints, which is usually the case. Contact LightPath Technologies for more detailed information on designing with GRADIUM capping.

All GRADIUM profiles are defined in pairs; there is a "positive" and "negative" profile. These are identical profiles that are reversed in the axial direction. When modeling lenses in double pass, it is necessary to use one profile for the forward pass, and the other (reversed) profile for the backward pass.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIUM SURFACES

Parameter 0	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
boule thickness	Δt	Δz	n_{ref}	d_x	d_y	t_x	t_y	Capping

Gradient 9

The Gradient 9 surface may be used to model SELFOC ® materials available from NSG America, Inc., or any gradient glass with a similar index variation. The shape of the Gradient 9 surface is the same as the Standard surface plus a "tilt" term in both x and y directions:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + x\tan(\alpha) + y\tan(\beta),$$

where c is the curvature (the reciprocal of the radius), r is the radial coordinate in lens units, k is the conic constant, and tan α and tan β are the tangents of the tilt angles in x and y. Note this is not the same surface shape as a tilted Standard surface, but it is a close approximation as long as the curvature is small, or if the tilt angles are small. The Gradient 9 surface has the following gradient profile:

$$n = n_0 \left[1.0 - \frac{A}{2} r^2 \right].$$

Both A and n_0 are functions of wavelength:

$$A(\lambda) = \left[K_0 + \frac{K_1}{\lambda^2} + \frac{K_2}{\lambda^4} \right]^2,$$

$$n_0 = B + \frac{C}{\lambda^2},$$

where the wavelength is measured in micrometers. The dispersion data is user defined and is stored in an ASCII file called GRADIENT_9.DAT. This file is stored in the folder <glass> (see "Folders"). The GRADIENT_9.DAT file contains blocks of 6 lines each. The first line in the file is the name of the material, which can be any name (without special characters such as spaces or quotes) less than 20 characters long. The next 5 lines are the values for B, C, K0, K1, and K2. There are no blank lines allowed between blocks. Zemax can read in data for up to 25 different materials in the GRADIENT_9.DAT file.

The dispersion data in the supplied GRADIENT_9.DAT file was provided by NSG America, Inc., in Somerset, NJ, (732) 469-9650, or by Corning, Inc., Corning NY, (607) 974-4755. Contact the respective manufacturers for detailed information on material properties. Not all of the materials offered by these manufacturers are necessarily included in the

GRADIENT_9.DAT file. The included materials are: SLS-1.0, SLS-2.0, SLW-1.0, SLW-1.8, SLW-2.0, SLW-3.0, SLW-4.0, and SLH-1.8 for NSG and Corning-Grin for Corning.

To use the Gradient 9 surface materials, change the surface type to Gradient 9, and then enter the appropriate material name in the glass column of the Lens Data Editor.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 9 SURFACES

Parameter 1	Parameter 2	Parameter 3
Δt	$\tan \alpha$	$\tan \beta$

Gradient 10

This surface has the same shape as the Standard surface, with media whose index of refraction is described by

$$ n = n_0 + n_{y1}y_a + n_{y2}y_a{}^2 + n_{y3}y_a{}^3 + n_{y4}y_a{}^4 + n_{y5}y_a{}^5 + n_{y6}y_a{}^6, $$

where $y_a = |y|$ and the || symbols indicate the absolute value. This form of gradient has a discontinuity at the plane $y = 0$, and the gradient is bisymmetric about the $y = 0$ plane. If the glass type is left blank, then there is no dispersion. If a glass type is entered in the glass column, then it must be one of the Gradient 5 materials defined in the section on the Gradient 5 surface type. The above formula then defines the index profile at the reference wavelength for the material, and the index at other wavelengths is computed according to the dispersion model defined for the Gradient 5 surface. The linear transverse term, n_{y1}, is ignored when performing paraxial ray tracing.

The maximum step size Δt determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces". Also see "Restrictions on surfaces following GRIN surfaces".

PARAMETER DEFINITIONS FOR GRADIENT 10 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Δt	n_0	n_{y1}	n_{y2}	n_{y3}	n_{y4}	n_{y5}	n_{y6}

Grid Gradient

This surface is a flat circular surface with an index of refraction defined for a grid of coordinates in three dimensions. The index data is user defined and is stored in any ASCII file with a .GGD extension. Select the file name from the

"GGD File" control of the Type tab of the Surface Properties dialog (see "Surface Type"). The index data file must reside in the <glass> folder (see "Folders"). Zemax will interpolate between the points defined in the GGD file using a tri-cubic spline.

GGD File Format

The file is split into two parts: the first line is the header and the second is the index and dispersion data. The header contains information about the contents of the file and the dispersion formula that Zemax will use. The main part will contain either index data, dispersion coefficients or both. The syntax is shown below.

```
DISPERSION_FORMULA_IDENTIFIER NX NY NZ DX DY DZ X Y Z n0 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10
.
.
.
```

In the header, the dispersion formula is a string that corresponds to one of the dispersion formulas described under "The glass dispersion formulas". The dispersion formula identifiers are the formula names with the spaces removed (e.g. the Sellmeier 3 formula would be Sellmeier3). Only those formulas listed in the aforementioned section are currently supported. The NX, NY and NZ values are the number of x, y and z points in the file, respectively. These values must be greater than 5. The DX, DY and DZ are the spacing between the points in x, y and z, respectively.

In the main body of the file the first three values are the x y and z coordinates. These should be evenly spaced points that increase monotonically over their range. That is, z should increase from zero to its maximum extent before repeating for the next set of x and y coordinates. The first index value is n_0, which is an index offset that can be set to zero if it is not required. The remaining values are dispersion coefficients that can be used to define an index distribution through the selected dispersion formula. For example, if you wished to define a Schott dispersion formula you would use the first five coefficients C1 through C5 and define the rest as zero. You would also have to make sure that Schott is the dispersion formula identifier defined in the file header. If you wish to define an index using the offset only, select a dispersion formula without a unity term, for example the Schott or Sellmeier 3 formulas, then set all the dispersion coefficients to zero.

The maximum step size is the only parameter for this surface type. See "Discussion on maximum step size for GRIN surfaces" for details.

PARAMETER DEFINITIONS FOR GRID GRADIENT SURFACES

Parameter 1
Δt

Grid Phase

This surface is nearly identical to the Grid Sag surface (see "Grid Sag"). The key differences are:

- The units of sag are radians of phase instead of units of length.

- The unitflag data is only used to scale the delx, dely, and derivative values.

- The surface shape is a plane.

- A diffraction order is supported. The diffraction order is a multiplier on the phase values. A value of zero will turn "off" the phase effects. Setting the order to -1 will reverse the sign of all defined phase values.

- A "shear distance" is supported. See "Using the shear distance" below.

The file format and general information provided in the grid sag description is otherwise valid.

Using the shear distance

One method of modeling atmospheric turbulence is to use a Grid Phase surface with data generated by an external atmospheric modeling program. Accurate modeling may require multiple surfaces with different phase data modeling atmospheric layers separated by long distances. Because of the separation between these layers, different field angles pass through different parts of the atmosphere, depending upon the distance between the layers and the ray angle. The separation causes a field dependent lateral shearing between the surfaces.

To model this shearing, one could in principle place multiple Grid Phase surfaces separated by large distances in front of the optical system. In practice, this works poorly, because the bending of the rays as they pass through a Grid Phase surface may cause the rays to miss the entrance pupil entirely. To avoid this problem, the Grid Phase surface may be placed at the entrance pupil of the system, and then the "shear distance" corresponding to the actual position of the phase layer may be defined. When ray tracing to the Grid Phase surface, for phase purposes only, Zemax will adjust the ray coordinates to correspond to the coordinates the incoming ray would have had if the Grid Phase surface had been placed at the shear distance. Normally, shear distances are negative as the distance is measured from the surface to the effective position of the phase layer. If the shear distance is zero, no lateral shifting occurs.

Phase coefficients sign conventions

See "Binary Optic 1" for a discussion of sign conventions.

Parameter 0	Parameter 1	Parameter 2
Diffraction order	Shear Distance	Interpolation method. Use 0 for bicubic spline, 1 for linear. See "Bicubic spline vs. linear interpolation" on page 354.

Grid Sag

The grid sag surface has a shape defined by a base plane, sphere, conic asphere, polynomial asphere, or Zernike asphere plus additional sag terms defined by a rectangular array of sag values. The surface shape is determined by either a linear or a bicubic spline interpolation of the sag values. The sag can be described at discrete points by

$$Z_{ij} = Z_{base} + Z(x_i - xdec, y_j - ydec),$$ where

$$z_{base} = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=1}^{8} \alpha_i r^{2i} + \sum_{i=1}^{N} A_i Z_i(\rho, \varphi),$$

$$1 \le i \le nx, \ 1 \le j \le ny,$$

and the Z_i terms are the Zernike standard terms as described in "Zernike Standard Sag". The terms delx and dely are the spacings between grid points in the x and y directions, xdec and ydec are optional decentration coordinates, and nx and ny are the number of points in the grid in each direction. The nx and ny values must be no smaller than 5. The terms defining Z_{base} are identical to those defined for the even asphere surface. Note that the sag defined by the grid points may be decentered relative to the sag defined by the base aspheric shape. There are also Zernike decenter terms in parameters 9 and 10, which are optionally used to decenter just the Zernike expression relative to the surface vertex. Zemax computes the effective paraxial curvature using the vertex radius of curvature, asphere coefficient data, and grid data points when computing paraxial power for this surface.

Importing grid data

All of this data must be computed and tabulated outside of Zemax, arranged in the proper file format, and then read using the Import section of the Surface Properties (see "Extra Data"). The proper file format is as follows:

```
nx ny delx dely unitflag xdec ydec
z dz/dx dz/dy d2z/dxdy nodata
.
.
.
```

The first line in the file contains seven values, which define the (integer) number of points in the x and y directions, the (floating point) increment in the x and y directions, an (integer) flag indicating the units of the data: 0 for mm, 1 for cm, 2 for in, and 3 for meters, and the (floating point) decenter of the grid points relative to the base surface in x and y. Any required scaling to the current lens units will be performed. Note that sag and cross derivative values have dimensions, and are therefore scaled, but first derivative values are dimensionless, and are not scaled.

The remaining nx*ny lines of the file contain four (floating point) numbers and (optionally) one integer each. The four floating point values are the sag, the x derivative of the sag, the y derivative of the sag, and the cross derivative d/dxdy. The optional fifth data entry is an integer flag that indicates if the data is invalid. Valid measured data should either have a zero or blank space for the nodata flag. Points for which the data is not valid should have a nodata value of unity. Currently, Zemax reads in the nodata flag value but does not use it; therefore, the sag data and derivatives should be set to zero or some other suitable value for points with no valid data.

The first data line in the file corresponds to the upper left corner of the surface, that is, the corner defined by -x and +y limits. Each point that follows is read across the face of the surface from left to right. After nx points, the nx+1 point is read in as the first value in row 2, and so on until nx*ny points have been read in. The file must be in ASCII and end in the extension .DAT (for files used by NSC objects, the extension should be .GRD).

The derivative values are required for smooth bicubic interpolation of the sag between the data points. The derivative values are not used by the linear interpolation algorithm. If all of the derivative values (dz/dx, dz/dy, and d2z/dxdy) are zero for every point in the file, then Zemax will automatically estimate the derivatives using a finite difference method.

The grid sag file format also supports comment lines. Any line starting with the "!" character is ignored.

Bicubic spline vs. linear interpolation

The sag of the surface at any arbitrary point is computed using either of two interpolation algorithms. The bicubic spline algorithm is used by default, and provides an interpolation that is smooth in both sag and the first derivative of the sag. The bicubic spline is the preferred algorithm for reasonably smooth data. For noisy pseudo- random like data, or grid data with sharp discontinuities, bicubic spline can cause "overshoot", which means the interpolation of the surface data can yield sag points which are far from the grid points. For these cases, a linear interpolation algorithm may provide more useful results. The choice of bicubic spline or linear interpolation is user selectable as one of the parameters.

Suggestions for using the Grid Sag surface

For important information about the limitations of splines, see "Comments about spline surfaces". To use the grid sag surface, first change the surface type to grid sag. Then go to the Import section of the Surface Properties dialog, and select the DAT file name. The data will be loaded into memory, but will not be displayed on the Lens Data Editor. The reason for not displaying the data is the potentially large number of points. Once the grid file is loaded, the values can be checked using the Surface Data Report. The data is thereafter stored with the lens file, and the original DAT file is ignored. If the DAT file must be changed, the file needs to be reloaded using the Import feature in the Surface Properties dialog.

The grid size is only limited by available memory. Each point in the grid requires 4 8-byte double precision values and 1 1-byte value, or 33 bytes. A 255 by 255 grid file would require approximately 2 Megabytes of memory. Surfaces using Grid Sag surfaces with many points can be slow to edit and read and write from files. One way to speed up editing is to turn off the "disk multi step" undo feature in Zemax. See the "Editors" section of "Project Preferences" (from the Setup Tab).

The bicubic interpolation algorithm is smooth to third order, with exact results at the grid points, and large grid files are generally not required for reasonably smooth surface shapes. However, like all low order spline models of surfaces, the Grid Sag surface can never closely model a high order asphere surface. The reason is that no reasonable number of piece-wise third order polynomials can accurately follow a higher order shape. The Grid Sag surface is intended to model arbitrary low order shapes without higher order undulations.

The Grid Sag surface is not defined outside of the bounds of the grid. Rays traced outside the grid area are treated as a ray miss error. It is a good idea to make the valid data portion of the grid slightly larger than the maximum area illuminated by rays; specifically, do not define the grid to be exactly the same size as the beam print. Rays nearly exactly at the edge of the grid may not trace as intended.

For Grid Sag surfaces that model steeply curved optics, such as fast mirrors, it is better to use the base radius and conic to define the parent shape, and then use the Grid Sag data to define the deviations from the base shape; rather than to embed the entire sag definition for the surface in the grid data. The reason is Zemax can use the base radius and conic to find a good "first guess" to the ray-surface intercept, then use the additional grid sag data to iterate to the exact solution. It is difficult and slow for Zemax to find the correct intercept between a steeply incident ray and a steeply curved Grid Sag surface if the base spherical surface is flat.

An example C language program, GRIDSAMP.C, can be found in the folder <objects>\Grid Files (see "Folders"). This sample program shows the correct method for creating a grid DAT file; the example used is a spherical surface. The GRIDSAMP program also shows how to use finite differences to compute the derivatives when only the sag formula of a surface is known analytically.

No optimization or tolerancing of grid data is possible.

PARAMETER DEFINITIONS FOR GRID SAG SURFACES

Parameter 0	Parameter 1-8	Parameter 9	Parameter 10
Interpolation method. Use 0 for bicubic spline, 1 for linear. See "Bicubic spline vs. linear interpolation" on page 354.	α_1 - α_8, see "Even Asphere" on page 336	Zernike decenter X in lens units	Zernike decenter Y in lens units

EXTRA DATA DEFINITIONS FOR GRID SAG SURFACES

Parameter Number	Description
11 - 12	Unused
13	Number of terms.
14	Normalization radius. Coordinates are normalized by this value.
15 - 245	Coefficients on Zernike polynomials 1 - 231, respectively, in lens units.

Hologram 1

The hologram 1 surface can be used to model optically constructed holograms. The hologram surface can be plane, spherical, or conical, and the medium behind the hologram can be air or glass. The glass can also be "MIRROR" which indicates the hologram is constructed and used in reflection. The hologram itself is described by the x, y, and z coordinates of two different construction points, a construction wavelength, and the diffraction order. The hologram deviates ray paths according to the equation

$$\hat{n} \times (r'_o - r'_r) = \left(\frac{m\lambda_p}{\lambda_c}\right)\hat{n} \times (r_o - r_r),$$

where $n^{\hat{}}$ is the unit vector normal to the surface of the hologram at ray intersection point, r_o is the unit vector along the first construction beam, r_r is the unit vector along the second construction beam, r'_r is the unit vector along the incident readout beam, r'_o is the refracted ray, λ_c and λ_p are the construction and playback wavelengths, and m is the diffraction order. A value of $m = 0$ means the ray is undeviated, while other integral values of m refer to higher diffracted orders. The notation used here is from the book "Aberrations of Optical Systems" by Welford, Adam Hilger (1986). Modeling holograms requires an understanding of their behavior which is beyond the scope of this section, and the user is advised to see the discussion in Welford, or some other reference, before using this feature.

Most holograms are constructed and used in transmission or reflection. There are occasions where the hologram is constructed in transmission, and then the substrate is aluminized and used in reflection. This special case can be simulated with the hologram surface by specifying a negative construction wavelength. Although ray tracing will be correct for this special case, OPD tracing will not work.

Zemax only models holograms to the extent of deviating ray paths. Other properties, such as efficiency and relative transmission are not supported.

The two construction beams are defined in terms of their source points. The x, y, and z coordinates of the source points are measured relative to the hologram vertex coordinate are defined in current lens units. Zemax computes the unit vector at the ray-surface intersection point using the local coordinate data and the construction point data for the two construction beams. The construction wavelength always has units of micrometers. If the distance to a source point is greater than 1.0E+08 lens units, the point is assumed to be infinitely distant and the construction beam is a perfect plane wave for that point. This assumption yields more accurate OPD computations than simply entering large coordinate values.

The hologram is defined by the interference between the two defined construction beams, with no aberrations assumed on the construction beams. Optically fabricated holograms with aberrated construction beams may be modeled in a very general way, see "Optically Fabricated Hologram".

Optically fabricated holograms with aberrated construction beams may be modeled using the "Optically Fabricated Hologram" surface.

The hologram 1 surface assumes that both construction beams diverge from the specified construction points. Because of the reciprocity of the construction beams, this is identical to the case where both construction beams converge toward the construction points. Some hologram fabrication methods require one beam to be converging while the other beam is diverging. See "The hologram 2 surface" for information on this latter type of hologram.

PARAMETER DEFINITIONS FOR HOLOGRAM SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
X_1	Y_1	Z_1	X_2	Y_2	Z_2	λ_c	m

Hologram 2

The hologram 2 surface is very similar to the hologram 1 surface. The key difference is that the hologram 1 surface assumes both construction beams either diverge from or converge to the construction points, whereas the hologram 2 surface assumes one construction beam converges to one construction point, and the other construction beam diverges from the other construction point. Which beam is first or second does not matter due to reciprocity. The parameter data is the same for both hologram 1 and hologram 2 surfaces.

Irregular

The irregular surface is a Standard surface shape (plane, spherical, or conic) that has additional aspheric deviations in terms of decenter, tilt, spherical aberration, astigmatism, and coma. This surface type is primarily used by the tolerancing algorithm to model irregularities in a standard shape surface. The surface sag is given by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + Z_s \rho^4 + Z_a \rho'^2_y + Z_c \rho^2 \rho'_y,$$

where

$$\rho_x = \frac{x}{r_{max}}, \rho_y = \frac{y}{r_{max}}, \rho = \sqrt{\rho_x^2 + \rho_y^2}, \rho'_y = \rho_y \cos\theta - \rho_x \sin\theta,$$

and r_{max} is the maximum radial aperture of the lens, defined by the semi-diameter value for the surface. The coefficients Z_s, Z_a, and Z_c represent the amount of spherical aberration, astigmatism, and coma, respectively, in lens units at the maximum radial aperture. The astigmatism and coma are oriented along a line that makes an angle θ in degrees with respect to the y axis.

The x and y coordinates of the previous equations are in a decentered and tilted coordinate system defined by the decenter x, decenter y, tilt about x, and tilt about y values. The decenters are in lens units, and the tilt is in degrees. The tilt and decenter values work exactly like the coordinate break surface defined in this section, however, the tilts and decenters are undone after the ray is traced to the surface. Ray tracing is done according to this algorithm:

The surface is decentered, tilted about x, then about y.

The ray is traced to the surface.

The surface is untilted about y, untilted about x, then undecentered.

The irregular surface uses the first seven parameters to define the decenter, tilt, and Z coefficients, and the eighth parameter to define the angle. All the coefficients are measured in lens units, except the tilt angles which are in degrees.

PARAMETER DEFINITIONS FOR IRREGULAR SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Decenter X	Decenter Y	Tilt About X	Tilt About Y	Zs	Za	Zc	θ

Jones Matrix

This surface is used to define an arbitrary polarizing component. The surface shape is always plane. The Jones matrix modifies a Jones vector (which describes the electric field) according to:

$$\begin{bmatrix} E'_x \\ E'_y \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} E_x \\ E_y \end{bmatrix},$$

where A, B, C, D, Ex, and Ey are all complex numbers. See "Defining polarizing components" for a complete discussion of the Jones matrix. Only polarization analysis features consider the effects of this surface type. This surface uses eight parameter values, and none of the extra data values.

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Ar	Ai	Br	Bi	Cr	Ci	Dr	Di

Lenset Array

Lenslet arrays are modeled using the User Defined Surface described in "Lenslet arrays using DLLs".

Non-Sequential Components

The sag of the Non-Sequential Components surface is the same as the Standard surface.

This surface enables ray tracing through one or more non-sequential objects. See "Non-Sequential Overview" for details.

Odd Asphere

The odd asphere model deviation is similar to the even asphere, except both even and odd powers of r are used. The sag is given by

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \beta_1 r^1 + \beta_2 r^2 + \beta_3 r^3 + \beta_4 r^4 + \beta_5 r^5 + \beta_6 r^6 + \beta_7 r^7 + \beta_8 r^8.$$

Note that the coefficients have units. The coefficients are entered in the corresponding parameter columns, as shown in the following table.

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
β_1	β_2	β_3	β_4	β_5	β_6	β_7	β_8

Odd Cosine

The odd cosine surface is an extension of the Odd Asphere surface, with 16 radial terms plus up to 6 additional cosine terms. The sag is given by:

$$z = \frac{cr^2}{1+\sqrt{1-(1+k)c^2r^2}} + \sum_{i=1}^{16}\beta_i r^i + \sum_{i=1}^{m} A_i s^{P_i}\cos(B_i\theta + C_i).$$

The first term is the sag for a Standard surface (plane, sphere, or conic). The second term is similar to the Odd Asphere surface, but the number of coefficients is fixed at 16. The third term supports m cosine terms, where the integer m must be between 0 and 6, inclusive. The coordinate s is the normalized radial coordinate given by s = r / R, where R is the user defined normalization radius. If R is zero or negative, the cosine terms are ignored. The surface may not be continuous in the first derivative if any B values is not an integer. The β_i coefficients have units which depend upon the index i, and the A_i coefficients have units of length in lens units. The angle θ is measured in radians and is related to the x and y coordinates on the surface by: tan θ = y/x.

The coefficient B is dimensionless and C is in units of radians. The coefficients are entered in the corresponding extra data columns, as shown in the following table.

EXTRA DATA DEFINITIONS FOR ODD COSINE SURFACES

Parameter Number	Description
13 - 28	The odd aspheric coefficients β_i
29	The number of cosine terms m, must be between 0 and 6.
30	The normalization radius R for the coordinate s in the cosine portion of the sag formula
31 - 54	The A, P, B, and C terms in the cosine portion of the sag formula.

Optically Fabricated Hologram

The Hologram 1, Hologram 2, and Toroidal Hologram surfaces described elsewhere in this file are used to define optically fabricated holograms assuming the construction optics are "perfect" and no aberrations are imposed on the construction beams.

The Optically Fabricated Hologram is far more general. The hologram is defined using 3 Zemax lens files:

1) The playback file, in which the Optically Fabricated Hologram surface is placed.

2) The construction file for beam #1

3) The construction file for beam #2

When rays are traced in the playback system, Zemax automatically calls the two construction systems to determine the vectors defining the interference of the two construction beams at the interception point in the playback system. There are significant advantages to this method of defining the hologram:

1) The construction optics may be completely arbitrary. Each construction file may consist of multiple lenses, mirrors, or even other holograms; anything Zemax can model. The aberrations introduced in the construction beams are therefore fully considered.

2) The two construction optics files may include completely different optics.

3) Any variables set in the construction system automatically become variable in the playback system; allowing simultaneous closed loop optimization of the construction and playback systems.

Defining the substrate shape

The shape of the Optically Fabricated Hologram surface substrate can be any of three possible shapes, defined by the "Shape" parameter zero:

Shape 0: Same as the "Standard", using c and k.

Shape 1: Same as the "Elliptical Grating surface shape", using a, b, c, and the polynomial terms.

Shape 2: Same as the "Toroidal Grating", using c, k, and R.

Defining the construction optics

To define an Optically Fabricated Hologram, first create two lens files that each define the optics used to illuminate and record the hologram. There are several important rules which MUST be followed when defining the construction optics files:

1) The files must have the same name, with a "_1" and "_2" appended to the end. For example, the file names "FAB_1" and "FAB_2" are valid file names for a pair construction optics files.

2) The construction files must reside in the same folder as the playback lens file.

3) The surface at which the two construction beams will interfere must be the stop surface in each file. Only the ray intercept vectors at the respective stop surfaces will determine the hologram properties. The stop surface may be a different surface number in each file.

4) The stop surface in each construction file must be either standard, elliptical grating, or toroidal grating surface types.

5) Only 1 configuration is allowed in each construction file. The playback system may use multiple configurations.

6) Only 1 field point is allowed in each construction file, although the field definition is arbitrary. Vignetting factors should not be used.

7) Only 1 wavelength is allowed in each construction file, and the wavelengths in each construction file must be identical. The playback wavelength is arbitrary, and need not be the same as in the construction optics.

8) Ray aiming must be turned on in each construction file.

9) The system aperture type must be "float by stop size" in each construction file.

10) No virtual propagations are allowed in the construction files; the rays must all take physically significant paths.

With the construction optics files defined, the name of the files is given to Zemax via the comments column. In the playback system, the hologram surface is set to "Optically Fabricated Hologram" and the comment for that surface is set to the name that defines the construction optics file names. For example, if "FAB_1" and "FAB_2" are used for the construction optics file names, enter "FAB" in the comment column. Zemax will append the "_1" and "_2" and automatically read in the construction optics files. If no name is given, the surface will ignore the effects of the hologram.

Optimizing the construction and playback optics

As Zemax traces rays in the playback system, any variables which affect the shape of the playback hologram (the semi-diameter, a, b, and c parameters, or extra data parameters) are automatically copied into the construction optics stop surfaces. This is why the stop surfaces in the construction and playback optics must be of the same type, so the construction beams overlap on a surface with the same shape as the playback surface.

Any variables set in the construction optics automatically become variable in the playback system, so the construction optics may be optimized simultaneously with any and all variables in the playback system.

Whenever the playback lens file is saved; the construction optics files are automatically saved as well.

To constrain design parameters in the construction files while optimizing both the playback and construction files simultaneously, use the CMFV operand in the playback system's merit function to call the merit functions defined in each of the construction beams. See "Optimization" for information about the CMFV operand.

Because this surface uses external Zemax files to define the construction beams, the Global Optimization algorithm should not be used. The other optimization algorithms are acceptable to use.

Selecting the hologram type

There are two ways of interfering the hologram construction beams:

1) By assuming that both construction beams either diverge or converge from their respective sources, or

2) By assuming that one source beam converges and one diverges from their respective sources.

Assumption #1 is analogous to the model for the Hologram 1 surface, while assumption #2 is analogous to the Hologram 2 surface. The Optically Fabricated Hologram surface supports both modes of interference. To choose the hologram 1 type, set the "Holo Type" in parameter 1 to be 1, otherwise, set the "Holo Type" in parameter 1 to be 2. The type must be either 1 or 2.

In the majority of cases, Zemax can correctly compute the OPD through the hologram using the default "OPD Mode" of zero. However, there are some special cases where the OPD is not computed correctly. For these cases, the user must manually determine the proper OPD algorithm by setting the OPD Mode equal to 1, 2, 3, or 4. The correct OPD Mode is a function of the construction and playback geometry and the total number of mirrors in the three files. There is no reliable algorithm for determining the proper OPD Mode automatically in all cases. If the OPD is obviously wrong, try each of the OPD mode values 0 - 4 until the OPD values are computed correctly.

PARAMETER DEFINITIONS FOR OPTICALLY FABRICATED HOLOGRAM SURFACES

Parameter 0	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
Shape	Holo Type	Diffraction order	c, a, or c	k, b, or k	-, c, or R	OPD Mode

Paraxial

The paraxial surface acts as an ideal thin lens. Two parameters are required to model the paraxial surface: the focal length and the OPD mode. The focal length is that which would be measured in air (unity index) although the paraxial model will support imaging into a non-unity index medium. If the focal length is zero the paraxial lens has no optical power. The OPD mode indicates how Zemax should calculate the optical path difference for rays refracted by the paraxial lens. Although the ray tracing through paraxial lenses is well defined (see below), computing the OPD is more difficult, especially in the presence of significant aberrations, as discussed below.

Setting the OPD Mode

Mode = 0 is very fast and accurate in most cases if the aberrations are not severe. If the Mode = 0, then the OPD computation is based upon the conjugate positions computed by tracing a real parabasal ray. This is preferred for axial optical systems with modest aberrations.

Mode = 1 integrates the actual phase introduced by the surface for each ray traced. Mode = 1 is substantially slower than the other modes, but is generally the most accurate. If the aberrations in the optical spaces on either side of the paraxial lens are large, the paraxial lens OPD computation cannot assume that the lens is working at fixed conjugates for all incoming rays. For these systems, the OPD mode should be set to 1. Mode = 1 can also be used to check if one of the

other modes is accurate. If the same OPD results are produced by Mode = 1 as another mode, then the other mode may be used as it will generally compute the OPD much faster.

Mode = 2 assumes that the lens is used at infinite conjugates regardless of the actual conjugates. This option will return incorrect results if the incoming beam is not reasonably collimated.

Mode = 3 is similar to Mode = 0, except paraxial rather than real rays are used. Mode = 3 is only valid for systems with modest aberrations that are well described by first order optical theory.

For maximum OPD computation accuracy, the working F/# of the paraxial lens should be no faster than about F/4, regardless of the OPD mode selected.

> **To model systems with collimated output, consider using Afocal Image Space rather than a paraxial lens; see "Afocal Image Space"**

The paraxial surface refracts rays using the following equations:

$$n'u_x' = nu_x - x\phi$$

$$n'u_y' = nu_y - y\phi$$

where ϕ is the surface power, n is the index of refraction, primes indicate values on the image side of the surface, and the angles are slopes which are computed from the ray direction cosines:

$$u_x = \frac{l}{n}$$

$$u_y = \frac{m}{n}$$

The paraxial surface shape is plane.

PARAMETER DEFINITIONS FOR PARAXIAL SURFACES

Parameter 1	Parameter 2
Focal length	OPD Mode

Paraxial XY

The paraxial XY surface type is similar to the paraxial surface except the optical power can be specified in the X and Y directions separately. This surface can therefore be used as a paraxial cylindrical or toroidal lens. Two parameters are required to define the paraxial XY surface: the X power, and the Y power. OPD through paraxial XY surfaces is computed using mode = 1 described in the paraxial surface description. Optical power is measured in inverse length lens units, such as inverse millimeters if the lens units are millimeters. See "Units".

The paraxial XY surface shape is plane.

PARAMETER DEFINITIONS FOR PARAXIAL XY SURFACES

Parameter 1	Parameter 2
X Power	Y Power

Periodic

This surface shape is described by the following expression:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} - A\left[\frac{1}{4}[1 + \cos(2\pi\alpha x)][1 + \cos(2\pi\beta y)] - 1\right],$$

where A is the peak to valley height in lens units of the modulation, and the values α and β are the spatial frequencies of the oscillations in the x and y directions, respectively. Note this expression is the sag of a sphere plus the cosine modulation. The sag of the surface is exactly zero at the vertex, and the amplitude is the peak to valley amplitude. The frequencies are measured in inverse lens units.

This surface uses three parameter values, and none of the extra data values.

PARAMETER DEFINITIONS FOR PERIODIC SURFACES

Parameter 1	Parameter 2	Parameter 3
A	α	β

Polynomial

The base radius of curvature and the conic constant are not used by this surface model. The sag of the polynomial surface is given by

$$z = \gamma_1 x^2 + \gamma_2 x^4 + \gamma_3 x^6 + \gamma_4 x^8 + \gamma_5 y^2 + \gamma_6 y^4 + \gamma_7 y^6 + \gamma_8 y^8.$$

A more general polynomial surface is also available, see "Extended Polynomial".

PARAMETER DEFINITIONS FOR POLYNOMIAL SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
γ_1	γ_2	γ_3	γ_4	γ_5	γ_6	γ_7	γ_8

Q-Type Asphere

The Q-Type asphere is a radially symmetric asphere similar to the Extended Asphere surface (see "Extended Asphere") with an alternate polynomial formulation. This surface supports two different variations of the Q-Type asphere, commonly called Qbfs and Qcon. The surfaces are described in G. W. Forbes, "Robust, efficient computational methods for axially symmetric optical aspheres", Opt. Express 18, 19700-19712 (2010) and G. W. Forbes, "Manufacturability estimates for optical surfaces", Opt. Express, Vol. 19, 9923-9941 (2011). The mathematics of these surfaces are fairly involved compared to traditional aspheric polynomials and the details are not reproduced here.

The extra data terms this surface uses are defined in the following table.

EXTRA DATA DEFINITIONS FOR Q-TYPE ASPHERE SURFACES

Parameter Number	Description
13	Type. Use 0 for Qbfs and 1 for Qcon.
14	Maximum term number. The maximum is 100.
15	Normalization radius in lens units. This is used by the algorithm to normalize the radial coordinates, analogous to the Extended Asphere surface.
16	Coefficient A_0
n	Coefficient $A_{(n-16)}$, where n has a maximum given by the maximum term number.

Radial Grating

The radial grating surface is similar to the diffraction grating surface (see "Diffraction Grating"), except the grating lines have radial symmetry, the grating line spacing is variable over the surface, and the substrate is an even aspheric surface shape (see "Even Asphere"). For a plane grating, rays traced to the grating are refracted according to the equation

$$n_2 \sin\theta_2 - n_1 \sin\theta_1 = \frac{M\lambda}{d},$$

where d is the grating spacing (always in micrometers), θ_2 is the refracted angle, θ_1 is the incident angle, M is the diffraction order, λ is the wavelength (always in micrometers), and n_1 and n_2 are the indices of refraction before and after the grating. The radial grating surface allows d to vary over the surface according to the equation:

$$d(p) = A_0 + A_1 p^1 + A_2 p^{-1} + A_3 p^2 + A_4 p^{-2} + \ldots,$$

where the A_i coefficients all have units of micrometers, and p is the normalized radial coordinate defined by

$$p = \frac{r}{R},$$

where r is the radial coordinate on the surface in lens units and R is the user defined normalization radius. The grating spacing d can be interpreted in two different ways. The normal Zemax convention is to measure d along the projection of the grating on the XY plane, ignoring any sag or curvature of the underlying surface. The Radial Grating surface

supports an additional "Grating Mode", where d is interpreted as being measured along the local surface tangent. The Grating Mode can be set to 0 or 1 in parameter 9 in the Lens Data Editor.

Note that the sign convention for M is arbitrary. The grating surface can be plane, spherical, conical, or even aspheric, and the medium before the grating, as well as the grating itself, can be air, glass, "MIRROR" or any other valid glass type. Zemax only models gratings to the extent of deviating ray paths. Other properties, such as efficiency, and relative transmission are not supported. If the grating spacing is too small to satisfy the grating relation, then a "Ray missed surface" error will be reported.

PARAMETER DEFINITIONS FOR RADIAL GRATING SURFACES

Parameter 0	Parameter 1-8	Parameter 9
Diffraction Order	Even aspheric coefficients $\alpha_1 - \alpha_8$	Grating Mode

EXTRA DATA DEFINITIONS FOR RADIAL GRATING SURFACES

Parameter Number	Description
10 - 12	Unused
13	Maximum number of terms
14	Normalization Radius
15	Coefficient on p^0
16	Coefficient on p^1
n	Coefficient on $p^{(n-15)}$

Radial NURBS

This feature is only available in the Professional and Premium editions of Zemax.

The acronym NURBS stands for Non-Uniform Rational B-Spline. NURBS are a very general class of curves and surfaces. For a complete discussion of NURBS, which is well beyond the scope of this discussion, see The NURBS Book, Second Edition, by Les Piegl and Wayne Tiller, Springer-Verlag, ISBN 3-540-61545-8.

The radial NURBS surface is defined by a series of weighted control points. The control points define a curve which starts at the origin and lies in the YZ plane along the +Y direction. Once this curve is defined, a figure of revolution is formed by rotating the curve a full 360 degrees around the Z axis. Unlike a spline surface, a NURBS curve does not actually go through the control points, except for the first and last control points.

'Each control point has a positive "y" coordinate, a "z" coordinate (which may be positive or negative), and a weight "w". The first point is always at y = 0 and z = 0 (as this point cannot be modified by the user, it is defined as control point 0), so the curve begins at the vertex origin of the surface. Subsequent points, numbered from 1 to the maximum number of points desired, extend monotonically outward in the +Y direction. The y values must be spaced at least 1.0E-3 lens units apart for numerical stability in the spline fit. When using this surface to design an imaging system (i.e. one in which wavefront error is an important metric), the z coordinate for control point 1 should be set to zero and the y coordinate for control point 1 should be set to a small fraction of the y coordinate for control point 2 (in this case control point 2 is really the first point of interest on the curve); doing this will avoid discontinuities in the surface shape near the origin which can otherwise lead to erroneous results for the wavefront error.'

The weights should all be set to 1.0 initially. The higher the weight, the closer the curve will be to the actual control point. Lower weights mean the curve is less constrained to be near the control point. See the above reference for details on the influence of weights on a NURBS curve.

The y value of the last point defined determines the maximum radial clear aperture of the surface. This value should generally be fixed, and not be a variable. Any rays which do not intercept the surface within the defined areas of the curve will be terminated with a "ray miss" error.

The advantage to the NURBS description is that any shape may be defined and traced reliably. Unusual aspheric correctors which cannot be described by polynomials may be modeled as NURBS.

The disadvantages to the NURBS description include very slow ray tracing, and sometimes great difficulty in finding suitable starting values for the control curve points and weights.

This surface does not use any of the parameter columns.

EXTRA DATA DEFINITIONS FOR RADIAL NURBS SURFACES

Parameter Number	Description
13	Number of control points. At least 4 points are required, and no more than 60 allowed.
14	y coordinate for control point 1.
15	z coordinate (sag) for control point 1.
16	w (weight) value for control point 1.
3(n+4)-1, 3(n+4), 3(n+4)+1	y, z, and w for control point n.

Retro Reflect

The Retro Reflect surface is a planar shape surface. If the surface is made of glass or air, rays refract according to Snell's law in the usual way. If the surface is made of "MIRROR", rays do not reflect in the usual way; instead the rays are

undeviated. This means under subsequent propagation in the opposite direction the rays will follow the incident path; that is, the rays retro-reflect.

Zemax cannot accurately compute the optical path difference of rays retro-reflected from this surface if the aberrations are large (hundreds of times the diffraction limit) and the beam incident upon the retro-reflect surface is very fast (about F/2 or faster). For this reason, the retro reflect surface should only be used with reasonably slow beams and modest aberrations.

This surface uses no parameter or extra data values.

Slide

This feature is only available in the Professional and Premium editions of Zemax.

This surface models a planar bitmap image. The values of the individual pixels in the bitmap define the transmission of the surface. Select the file name from the "Bitmap Image" control of the Type tab of the Surface Properties dialog (see "Surface Type"). The bitmap image file must reside in the <images> folder (see "Folders").

If the Mode parameter is 0, rays outside of the bitmap image will have a transmission of 0. If Mode is 1, these rays will have a transmission of 1. The X-Half Width defines the half width of the bitmap in lens units. The bitmap is always centered at the vertex of the surface. The Y-Half Width is automatically determined from the aspect ratio of the bitmap itself, the pixels are assumed to be square. If the Aspect Ratio is greater than zero, then this value defines the height over width aspect ratio of the individual pixels in the bitmap. The default aspect ratio is 1.0.

The transmission of the slide is based upon the RGB values of each pixel and the incident wavelength. Because the RGB values do not correspond to a specific wavelength spectrum, a linear interpolation algorithm is used to estimate the bitmap transmission.

PARAMETER DEFINITIONS FOR SLIDE SURFACES

Parameter 1	Parameter 2	Parameter 3
Mode	X-Half Width	Aspect Ratio

Standard

The most commonly used optical surface is a spherical surface. The sphere is centered on the current optical axis, with the vertex located at the current axis position. Zemax treats planes as a special case of the sphere (a sphere with infinite radius of curvature) and conics as a special case as well. The "sag" or z-coordinate of the Standard surface is given by

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}}$$

where c is the curvature (the reciprocal of the radius), r is the radial coordinate in lens units and k is the conic constant. The conic constant is less than -1 for hyperbolas, -1 for parabolas, between -1 and 0 for ellipses, 0 for spheres, and greater than 0 for oblate ellipsoids. For more information on conic constants, see "REFERENCES ON LENS DESIGN". The Standard surface does not use any of the parameter values.

Modeling an ellipse with the Standard surface

There are a few handy formulas for converting the semi major and semi minor axis lengths of an elliptical surface to a radius and conic description. If "a" is the semi major axis length, and "b" is the semi minor axis length, then

$$\frac{1}{c} = R = \pm\frac{b^2}{a},$$

$$k = -\varepsilon^2 = -\left[\frac{a^2 - b^2}{a^2}\right].$$

Modeling an axicon with the Standard surface

The Standard surface can be used to make an almost perfect axicon. If $(1 + k)\, c^2\, r^2 \gg 1$, the Standard surface reduces to

$$z = \frac{r}{\sqrt{(-(1+k))}} \text{ or } z = r \tan\alpha, \text{ where } \tan\alpha = \frac{1}{\sqrt{(-(1+k))}},$$

and α is the axicon angle, measured from the XY plane to the axicon surface. To create an axicon, calculate the conic constant value (k) from the angle ( ) desired, and use any small value for the radius of curvature. The resulting value of k must be negative. The exact value of the radius or curvature is not important, as long as it is roughly three or more orders of magnitude smaller than the radial aperture of the axicon. The axicon is not perfect in the sense that there is no cusp at the origin; the region around the surface vertex will be rounded off over a size given approximately by the radius value. This is actually a desirable property for ray tracing, as the surface is everywhere smooth. However, paraxial data, such as effective focal length, magnification, and other common first-order optical properties are not generally meaningful for axicon surfaces. Even though the surface shape is still a conic asphere, the shape of the axicon implies the paraxial properties of the system are not representative of the optical surface as a whole. This is the case for any optical surface that is not well described by the radius of curvature alone.

Superconic

The most common form of a polynomial aspheric surface uses a power series expansion in the radial coordinate r to define the surface sag, where r is defined by

$$r^2 = x^2 + y^2.$$

For example, the Even Aspheric surface described in "Even Asphere" uses such an expansion. Since r does not depend upon z, the expansion term is the distance from the vertex to the point on the surface as projected on to the tangent plane. Generally, the departure of the asphere from the tangent plane increases with radial aperture. As the departure increases, the power series expansion parameter r corresponds to a point on the tangent plane which is farther from the point on the surface. This causes the expansion to have poor convergence.

A novel solution proposed by Alan Greynolds of Breault Research Organization is to instead expand in powers of the distance from the vertex to the point on the surface. The expansion is then in terms of

$$s^2 = x^2 + y^2 + z^2.$$

Starting with the conic equation for a surface

$$kz^2 - 2Rz + x^2 + y^2 + z^2 = 0,$$

where k is the conic constant and R is the radius of curvature, a general power series expansion can be made of the form

$$Az^2 - 2Bz + C = 0.$$

The constants are defined as

$$A = \frac{k}{R},$$

$$B = 1 + U_1 s^2 + U_2 s^4 + \ldots, \text{ and}$$

$$C = \frac{s^2}{R} + V_1 s^4 + V_2 s^6 + \ldots,$$

where U and V are coefficients which define the aspheric shape. Note that if all the U and V terms are zero, a standard conic results. If A is also zero, then the superconic becomes a sphere. The coefficients A, U_1, and V_1 together form a Cartesian oval. These properties make the superconic stable when optimizing for the coefficients. The superconic can be used to model surfaces which otherwise would require aspheric terms of very high order. Zemax models superconics with up to 240 terms, in practice designs rarely use more than 5 terms.

EXTRA DATA DEFINITIONS FOR SUPERCONIC SURFACES

Parameter Number	Description
13	Maximum term number. The maximum is 250, but 4-10 is typical.
14	U_1
15	V_1
Even n	$U_{(n - 12)/2}$

Odd n	$V_{(n - 13)/2}$

Tilted

The tilted surface is a plane that makes an angle with respect to the x and y axes. The surface is easily defined in terms of the tangent angle between the plane and the X and Y axes:

$$z = x\tan\theta_x + y\tan\theta_y.$$

The tilted surface uses the first two parameters to define the tangents of the x and y angles. This surface is very useful for implementing tilted object and image surfaces, as well as tilted faces on prisms. The tilted surface does not change the local coordinate system, and so should not be used for implementing fold mirrors; use the coordinate break surface instead.

PARAMETER DEFINITIONS FOR TILTED SURFACES

Parameter 1	Parameter 2
$\tan\theta_x$	$\tan\theta_y$

Toroidal

Toroidal surfaces are formed by defining a curve in the Y-Z plane, and then rotating this curve about an axis parallel to the Y axis and intersecting the Z axis. Toroids are defined using a base radius of curvature in the Y-Z plane, as well as a conic constant and polynomial aspheric coefficients. The curve in the Y-Z plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1 + k)c^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14}.$$

This curve is similar to the even aspheric surface sag formula, except the sixteenth order term has been omitted, and the coordinate argument is y, not . This curve is then rotated about an axis a distance R from the vertex. This distance R is referred to as the radius of rotation, and may be positive or negative. The Y-Z radius of curvature is specified in the same column on the spreadsheet editor as the radius for Standard surfaces. The radius of rotation is set on parameter column number 1. To model a cylinder lens which is flat in the X direction use zero, which Zemax interprets as infinite radius.

The surface may optionally be modified by the addition of the Zernike Standard sag terms:

$$z = z_t + \sum_{i = 1}^{N} A_i Z_i(\rho, \varphi),$$

Where z_t is the base toroidal sag and the Zernike Standard terms are defined in the section "Zernike Standard Sag", and are entered using the Import section of the Lens Data Editor. Note if the Y-Z radius is set to infinity, a surface with

power in x but not in y can be described, therefore, the cylinder may be oriented in either direction. The other parameter columns are used for the optional aspheric coefficients, as specified in the following table. If aspheric coefficients are required in the X direction, then rotate the toroid with a pair of coordinate break surfaces and rotate about Z. If different aspheric surfaces are required in both the X and Y directions, see the "biconic", "polynomial" and "extended polynomial" surfaces described elsewhere in this file.

PARAMETER DEFINITIONS FOR TOROIDAL SURFACES

Parameter 0	Parameter 1	Parameter 2-8
Extrapolate	Radius of Rotation	$\alpha_1 - \alpha_7$

EXTRA DATA DEFINITIONS FOR TOROIDAL SURFACES

Parameter Number	Description
13	Number of terms
14	Normalization radius. Coordinates are normalized by this value
15 - 245	Coefficients on Zernike polynomials 1 - 231, respectively, in lens units.

The "Number of terms" is used to specify the maximum Zernike polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation; terms beyond this number are ignored.

Zernike polynomials are orthogonal over the unit circle, and so the normalization radius should be set to the radius over which the coefficient data was normalized. Zernike polynomials tend to diverge quite rapidly beyond the normalization radius, and so care should be taken that rays do not strike the surface beyond this radius. Although the ray tracing algorithm may work, the data may be inaccurate. The extrapolate flag may be set to zero to ignore the Zernike terms for rays that land outside the normalization radius.

Toroidal Grating

Toroidal grating surfaces are similar to regular toroidal surfaces, except no aspheric sag terms are supported, and a diffraction grating may be placed on the toroidal surface. Toroidal gratings are described by defining a curve in the Y-Z plane, and then rotating this curve about an axis parallel to the Y axis and intersecting the Z axis. Toroidal gratings are defined using a base radius of curvature in the Y-Z plane, as well as a conic constant. The curve in the Y-Z plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1 + k)c^2 y^2}}.$$

This curve is similar to the Standard surface sag formula, except the coordinate argument is y, not r. This curve is then rotated about an axis a distance R from the vertex. This distance R is referred to as the radius of rotation, and may be positive or negative. The Y-Z radius of curvature is specified in the same column on the spreadsheet editor as the radius for Standard surfaces. The radius of rotation is set on parameter column number 1. To model a cylinder lens which is flat in the X direction use zero, which Zemax interprets as infinite radius.

Note if the Y-Z radius is set to infinity, a surface with power in x but not in y can be described, therefore, the cylinder may be oriented in either direction.

The diffraction grating is defined in terms of the number of lines per micrometer and the diffraction order. These values are specified in parameter columns 2 and 3, respectively. The grating lines are parallel to the local x axis, and are evenly spaced when projected onto a plane.

For a toroidal grating with aspheric deformation terms see "Extended Toroidal Grating".

PARAMETER DEFINITIONS FOR TOROIDAL GRATING SURFACES

Parameter 1	Parameter 2	Parameter 3
Radius of Rotation	Grating Lines per micrometer	Diffraction order

Toroidal Hologram

The Toroidal Hologram surface shape is identical to the Toroidal surface described in "Toroidal". The curve in the Y-Z plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1 + k)c^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14}.$$

The radius and conic values, along with the parameter values, are used to define the surface shape. The extra data values are used to define the holographic properties of the surface. The terms used to define the holographic construction beams are identical to those described in the earlier sections on the Hologram 1 and Hologram 2 surface types. Extra data values 1-6 are the x, y, and z construction points for sources 1 and 2, value 7 is the construction wavelength, and value 8 is the playback order M.

The hologram is defined by the interference between the two defined construction beams, with no aberrations assumed on the construction beams. Optically fabricated holograms with aberrated construction beams may be modeled in a very general way using the "Optically Fabricated Hologram".

> **Optically fabricated holograms with aberrated construction beams may be modeled using the "Optically Fabricated Hologram" surface.**

There is also a ninth extra data value which is used as a "flag" to indicate if the two construction beams are converging or diverging. If both beams are converging or diverging to or from the construction points, then the flag value should be +1. If one beam is converging and one is diverging, then the flag should be -1. Any value less than zero is the same as -1, and any value greater than zero is the same as +1. This flag value serves to distinguish these two cases, which is the only difference between the Hologram 1 and 2 surface types.

PARAMETER DEFINITIONS FOR TOROIDAL HOLOGRAM SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Radius of Rotation	α_1	α_2	α_3	α_4	α_5	α_6	α_7

EXTRA DATA DEFINITIONS FOR TOROIDAL HOLOGRAM SURFACES

Parameter Number	Description
13	Construction coordinate X1 for first construction beam.
14	Construction coordinate Y1 for first construction beam.

15	Construction coordinate Z1 for first construction beam.
16	Construction coordinate X2 for second construction beam.
17	Construction coordinate Y2 for second construction beam.
18	Construction coordinate Z2 for second construction beam.
19	Construction wavelength.
20	Playback order.
21	Converge/Diverge flag.

Toroidal NURBS

The toroidal NURBS surface is similar to the radial NURBS surface. The key difference is that instead of rotating the curve about the Z axis to form a surface; the curve is instead mirror-imaged to the -Y axis, then rotated about an offset Y axis to form a toric shape.

The offset axis may be in the +Z or -Z direction, depending upon the sign of the radius of rotation. If the radius of rotation is zero, then the radius is assumed to be infinite and the surface has cylindrical symmetry. The X width of the surface is determined by either the range of rotation angle or by direct specification of the -x and +x limits; the latter being used if the radius of rotation is infinite (which is defined by a zero value).

PARAMETER DEFINITIONS FOR TOROIDAL NURBS SURFACES

Parameter 1	Parameter 2	Parameter 3
Radius of Rotation	Min X or angle	Max X or angle

EXTRA DATA DEFINITIONS FOR TOROIDAL NURBS SURFACES

Parameter Number	Description
13	Number of control points. At least 4 points are required, and no more than 60 allowed.
14	y coordinate for control point 1.
15	z coordinate (sag) for control point 1.
16	w (weight) value for control point 1.
3(n+4)-1, 3(n+4), 3(n+4)+1	y, z, and w for control point n.

User Defined

This feature is only available in the Professional and Premium editions of Zemax.

The user defined surface (UDS) is a powerful, flexible, and fast way of implementing surfaces not already built into Zemax. A UDS may be any shape, have any refractive, reflective, or diffractive properties, may impart any arbitrary phase to the beam, and may be followed by either homogeneous media or by a gradient index media of arbitrary form. A UDS may also apodize or attenuate the beam arbitrarily, or be used to define any electric field or coating data for polarization analysis. This latter capability allows the beam to be partially transmitted according to an arbitrary formula or table as defined by the user at any surface in the optical system.

The secret to this great flexibility is that all the properties of the surface are defined by the user in a separate C or C++ program compiled and linked into Zemax dynamically using the Windows Dynamic Link Library (DLL) capability. The DLL must contain functions which return to Zemax all the data required to draw the surface, trace rays, compute refraction angles, and for gradient index, to determine the index as a function of position within the media following the surface.

Because the DLL is compiled code, and because Zemax passes to the DLL a pointer to the data that needs to be computed, UDS are very fast; nearly as fast as native Zemax code (there is a small amount of overhead due to the function call).

The power of the UDS comes at a price, although it is a reasonable one. Use of the UDS does require that the user have a suitable compiler or development tool that can generate 32 bit Windows compatible DLLs. It is also assumed that the user can write the required code, and most importantly, ensure that the code is reliable and bug-free. To maximize speed, Zemax performs very little error checking on data returned by the DLL, and so buggy UDS DLLs are quite capable of bringing Zemax to a crash.

For this reason, technical support on the implementation of UDS is strictly limited to illustrating that the provided sample files work correctly. If you need a UDS DLL, and do not possess the desire or ability to write them yourself, please feel free to contact Zemax Technical Support for a quote on developing a custom DLL to meet your requirements. We have considerable experience in developing ray tracing algorithms, and can generally write DLLs at very competitive rates on very short notice.

The UDS DLL

The best way to learn how to create a UDS DLL is to start by copying the source code file (the one that ends in .C) of a sample DLL to a new file name, edit the sample to suit your requirements, and then recompile to make a new DLL. A good portion of the DLL source code is "boiler plate" which is common to all DLLs.

There are two files that are required to define a UDS DLL: a C (or C++) source code file such as MY_SURF.C, and a header file called USERSURF.H. Only the C file needs to be modified. The C file defines 2 functions; the first is "DLLMain" which is used to (optionally) initialize data used by the DLL. The other function is "UserDefined-Surface4", and this is the function that is modified for each surface type. Earlier versions of the Zemax User Defined surface interface used the function names UserDefinedSurface, UserDefinedSurface2, and UserDefinedSurface3; these names are still supported but are obsolete. New DLL files should use UserDefinedSurface4.

Zemax passes two structures to the function, and these structures are defined in the USERSURF.H file. The USERSURF.H file defines a data structure used by UserDefinedSurface3 called "FIXED_DATA4". Inside the UserDefinedSurface4 function is a C "switch-case" construct. Zemax passes a "type" parameter, which is a number between 0 and 10, inclusive, that indicates what data Zemax needs computed and returned to Zemax. The type codes are described in the following table:

UDS DLL Type Codes

TYPE	What Zemax wants the DLL to compute
0	The surface name, radial symmetry status, and grin status
1	The name of the parameter columns
2	The name of the extra data columns
3	The sag and alternate sag of the surface
4	A paraxial ray trace to and refraction through the surface
5	A real ray trace to and refraction through the surface, including transmittance and polarization data if any
6	The index and first derivatives for gradient index propagation
7	The default data when the user first selects the surface type
8	The initialization of the DLL, if required. This includes allocating static memory the DLL will require, loading data files, or other one-time calculations or initializations. This type code is called once each time the DLL is loaded. However, Zemax may load the same DLL multiple times. For example, each analysis window in Zemax gets it's own copy of the lens data, and each copy will load and initialize the DLL every time the analysis window is updated. Zemax may update multiple analysis windows at the same time (parallel execution) and so multiple DLLs may be running simultaneously. For this reason, using fixed file names to store temporary data or to make log files is not recommended; the same file will be created or written to multiple times as Zemax loads the DLL multiple times. The DLL only needs to handle this type code if the DLL requires one-time initialization before the DLL can trace rays.
9	The termination of the DLL, if required. This includes releasing allocated memory or closing data files. This type value is called when the DLL is removed from the surface or the lens is closed. See the comments for type code 8 above for details on how Zemax may load, initialize, and terminate multiple DLLs at once. The DLL only needs to handle this type code if the DLL must release memory or perform some other "clean-up" before terminating.
10	The scaling of parameter and extra data values used by the DLL. The scale factor is stored in the USER_DATA structure in the path argument.

Extensive comments are provided in the sample DLL source code files. The DLL files can be found in the folder <program>\DLL\Surfaces, and any new DLL files must be placed there as well.

Refractive and reflective UDS DLLs

For conventional homogeneous surfaces which are reflective or refractive, but not gradient or diffractive, start with the file US_STAND.C. That file is the source code for US_STAND.DLL, which is a clone of the built-in Zemax Standard surface. It is not exactly the same code, and is somewhat slower than the form of equations used by Zemax, however, it is functionally equivalent. US_STAND.C includes a generic Snell's' law refraction function which also works for reflection. The DLL, like all UDS DLLs, provides the sag, ray intercept equations, surface normals, optical path, and paraxial refraction functionality.

Gradient index UDS DLLs

Gradient surfaces should use US_GRIN1.C as a starting point. The code is very similar to US_STAND.C, except a flag is returned indicating that the surface is a gradient type, and data type "6" is implemented to return the index of refraction given X, Y, Z, and all the parameter and extra data. The first derivative of the index in the X, Y, and Z directions must also be provided.

For gradient index UDS DLLs, the index of refraction used for paraxial ray tracing is the index at the front vertex of the surface, at the coordinates (0, 0, 0), as long as the glass column is left blank. If a glass type is specified for the DLL (in the usual glass column of the LDE) then the paraxial reference index is computed from the glass catalog data for that

material. It is not required, or even advisable, to specify a glass name when using gradient index DLLs. The only time a glass name should be provided is when the DLL computes an index offset based upon the catalog glass properties.

Diffractive UDS DLLs

Diffractive optics are very similar to standard optics, except the rays are further deviated by the derivative of the phase as a function of X and Y, and the optical path length needs to be modified to include the phase change.

Generally, "refraction" for diffractive optics is defined by:

$$l' = l + \frac{\lambda}{2\pi}\frac{\partial\varphi}{\partial x}$$

$$m' = m + \frac{\lambda}{2\pi}\frac{\partial\varphi}{\partial y}$$

where l and m are the direction cosines, and the z direction cosine, n, is computed to make the magnitude of the direction vector unity, and φ is the phase in radians.

A sample DLL, US_GRATE.DLL, illustrates the diffractive computation by cloning the Zemax grating surface.

Lenslet arrays using DLLs

Lenslet arrays are easily modeled using the user defined surface. Basically, the ray trace determines which segment of the array is struck, then uses the local lens curvature to determine the refraction. The sample source code and DLL are provided as US_ARRAY.C and US_ARRAY.DLL, respectively.

User defined surface apodization using DLLs

The real ray trace portion of the DLL, type code 5, allows definition of the surface transmittance. The surface transmittance must be a number between 0.0 and 1.0, and indicates the relative fraction of intensity that the ray transmits through the surface. In this context, transmit means "continues on", so for reflective surfaces, the transmitted portion is that which normally reflects to the next surface.

The surface transmittance can be used to define arbitrary surface apodizations. The transmittance function can be any formula based upon the ray coordinates, direction cosines, surface parameters, or other data; or may be derived from a look up table, or any other method that can be implemented within the DLL.

The surface transmittance need not be defined. If it is not defined by the DLL, Zemax assumes it is 1.0. Whatever the surface transmittance is, the ray intensity will still be modified by Zemax to account for the Fresnel surface, thin film, and bulk absorption affects that are normally accounted for when doing polarization ray tracing.

If Zemax is not doing polarization ray tracing, then the DLL defined transmittance is the only attenuating effect. Unlike pupil apodization, surface apodizations defined using UDS may be placed on any surface anywhere in the optical system. Note that for apodizations which are a function of ray position (and most are) different fields will then see a different effective apodization. Note that this technique can be used to model arbitrary neutral density filters, or filters whose transmission are wavelength dependent.

Polarization and coating data using DLLs

The real ray trace portion of the DLL, type code 5, allows definition of either the transmitted or reflected electric field directly, or the definition of the s and p orientation complex reflection and transmission coefficients. The field or coating data can be based upon any data available within the DLL, including ray cosines, normal vectors, index, or other user defined data, or any other method that can be implemented within the DLL. The field and coating data need not be

defined. If it is not defined by the DLL, Zemax uses the default algorithm for the surface. For a source code example of how to define the electric field or coating data see the sample DLL US_POLARIZATION.

Error handling and UDS

The convention Zemax uses internally is to return a zero value if the DLL computed a meaningful result, and no error occurred. Otherwise, the DLL should return -1. The exception is when ray tracing, either paraxial or real. If the ray misses the surface, it should return the surface number. If the ray total internally reflects (TIRs) then the return value should be the negative of the surface number. Zemax uses these error codes to help provide meaningful diagnostics to both the user and various Zemax features.

Sample DLLs

Numerous sample UDS DLLs have been written and are provided as both a ready to use DLL and as C source code for study and modification. The easiest way to write a DLL is to find one most similar to the one you need, and copy and edit the C source code file as required. The following table lists the DLLs available and a brief description of each. Note that although the sample DLLs have been tested and are generally considered reliable, they are provided "as is".

SAMPLE UDS DLL'S

DLL name	Description
QBFS, QCON	These two DLLs are an implementation of the surface types described in "Shape specification for axially symmetric optical surfaces", G. W. Forbes, Optics Express Vol. 15(8) pp 5218-5226 (April 16, 2007).
US_ANAMR	Anamorphic asphere. This aspheric surface is defined by the following expression: $$Z = \frac{CxX^2 + CyY^2}{1 + \sqrt{1 - (1 + Kx)(Cx^2X^2) - (1 + Ky)(Cy^2Y^2)}}$$ $$+ AR[(1 - AP)X^2 + (1 + AP)Y^2]^2 + BR[(1 - BP)X^2 + (1 + BP)Y^2]^3$$ $$+ CR[(1 - CP)X^2 + (1 + CP)Y^2]^4 + DR[(1 - DP)X^2 + (1 + DP)Y^2]^5$$
US_ANASP	Annular aspheric surface. This surface is described in the paper "Annular surfaces in annular field systems" by Jose Sasian, Opt. Eng. 36(12) 3401-3403 (December 1997). This surface is similar to an odd aspheric polynomial, with a radial offset in the polynomial expansion to better model surfaces used in annular field systems.
US_APGXY	This DLL is a clone of the Zemax "Standard" surface type, with the added capability to modify the surface transmission. This surface defines an apodization function of the form $$T(x, y) = Exp\left(-2G_x\left(\frac{x}{R}\right)^2 - 2G_y\left(\frac{y}{R}\right)^2\right),$$ where R is the semi-diameter of the surface and x and y are the coordinates at which the ray intercepts the surface.
US_ARRAY	Models an n x m array of lenses, each of dimension H x W; all 4 parameters may be specified. The lenses are rectangular in outline but are "standard" in shape; that is, plane, spherical, or conic asphere. The number of lenses in each direction must be odd.

DLL name	Description
US_ARRAYEVEN	Models an n x m array of lenses, each of dimension H x W; all 4 parameters may be specified. The lenses are rectangular in outline but are "even aspheric" in shape; that is, plane, spherical, conic asphere, and radial polynomial asphere. The number of lenses in each direction must be odd.
US_CYLAR	Models a vertical array of n cylindrical lenses, each H high. The lenses are cylindrical or conic aspheres in the YZ plane.
US_DGCYL	Models a diffraction grating on a cylindrical surface, where the grating lines are equally spaced along the arc of the surface rather than equally spaced along the y coordinate of the tangent plane. The grating lines are parallel to the axis of the cylinder.
US_DATASURFACE	This surface utilizes the FIXED_DATA4 structure and the ability to pass values from prior Data surfaces. Upon first load it will read extra data values from prior Data surfaces, indicate how many data values were read in Num Values parameter column, and copy those values into the EDE. It does not dynamically update. It is solely for illustrating the access to these values in the DLL.
US_EAPERIODIC	This surface is similar to the Even Asphere, with an additional term allowing for a sinusoidal variation of the sag. Such variation may be useful for modeling surface irregularities. The full sag expression for the surface is given by: $$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \sum_{i=1}^{8} \alpha_i \rho^{2i} - \frac{A}{2}[\cos(2\pi\omega_0 r + \phi_0) - \cos(\phi_0)]$$ where the parameters c, k, and α_i are the same as for the Even Asphere surface. The parameter A represents the magnitude of the sinusoidal sag perturbation, and should be specified in lens units. The parameter ω_0 represents the frequency of the sinusoidal sag perturbation, and should be specified in inverse lens units. The parameter ϕ_0 is the initial phase of the sinusoidal sag perturbation, and should be specified in degrees (the value is converted to radians inside of the DLL for calculation of the sag).
US_FILT1	This surface defines an apodization function of the form $T = 1.0 - Exp(-G\rho^2)$ where p is the radial coordinate normalized to the semi-diameter of the surface.
US_FILT2	This surface defines an apodization function where the optical density is Dmax from zero to a normalized radial coordinate p1, then linearly goes to D=0 at p2, and D= 0 thereafter. The density is D = -Log10(T) where T is the transmission.
US_FILT3	This surface defines an apodization function of the form $D = 0.5 D_{max}(1.0 + \cos\pi\rho)$ where p is the radial coordinate normalized to the semi-diameter of the surface.

DLL name	Description
US_FILT4	This surface has an apodization function defined by: $$T(r) = 1 \text{ if } r < R - \Delta R,$$ $$T(r) = 0 \text{ if } r > R + \Delta R,$$ $$T(r) = \frac{1 + \cos\left(\frac{\pi(r - (R - \Delta R))}{2\Delta R}\right)}{2} \text{ if } R - \Delta R < r < R + \Delta R,$$ where r is the radial coordinate, R is the aperture radius, and Δ is a dimensionless fraction of the aperture radius. The transmission varies smoothly from 1 to 0 over twice the distance defined by ΔR. The value for Δ is defined as 0.001 if the provided value for Δ is zero. This function makes a convenient "soft-edged" aperture which is useful for optimization of transmission through a circular aperture.
US_FILT5	This surface has an apodization function defined by: $$T(na) = 1.0 \text{ if na} < \text{NA} - \Delta,$$ $$T(na) = 0.0 \text{ if na} > \text{NA} + \Delta, \text{ otherwise}$$ $$T(na) = 0.5\left(1 + \cos\left(\frac{(na - (NA - \Delta NA))\pi}{(2\Delta NA)}\right)\right),$$ where na is the numerical aperture of the incoming ray, NA is a user defined parameter, and Δ is fixed at 0.001 (although this value may be edited in the DLL source code). This type of filter demonstrates apodization of a ray based upon it's incident angle, rather than position. Rays whose na exceeds the NA-Δ value will be attenuated smoothly to zero. This permits easier, more continuous optimization.
US_FILT6	This surface has an apodization function defined by: $$T(x, y) = (1/4)(1 + \sin(2\pi\alpha x))(1 + \sin(2\pi\beta y)),$$ where α and β are the spatial frequencies of the sinusoidal transmission in x and y, respectively.
US_FILT7	This surface apodizes the surface transmission via linear interpolation of values specified on the extra data editor as a function of either the radial coordinate on the surface or as a function of the angle the ray makes with respect to the local Z axis after refraction from the surface.
US_FILT8	This surface models a soft edged rectangular aperture. The MaxX and MaxY parameters are the X and Y half widths of the rectangular aperture. The transmission varies smoothly from 1 to 0 over twice the distance defined by the DelX and DelY distances from the aperture boundaries. This function makes a convenient "soft-edged" aperture which is useful for optimization of transmission through a rectangular aperture.

DLL name	Description
US_GCYL	Models a cylindrical surface with a gradient index medium. The GRIN medium has concentric cylindrical shells of constant index, and a polynomial GRIN profile centered on the center of curvature. This DLL was intended to model light transversing a fiber perpendicular to the axis of the fiber (through the side).
US_GRATE	Models a Standard surface with grating lines parallel to the x axis, equally spaced along y in the tangent plane. This is a clone of the Zemax "Diffraction Grating" surface. The purpose is to show how diffractive surfaces are modeled in Zemax UDS DLLs.
US_GRIN1	Models a quadratic GRIN medium. This is a clone of the Zemax "Gradient 1" surface. The purpose is to show how GRIN surfaces are modeled in Zemax UDS DLLs.
US_IGRIN	An example of a GRIN with dispersion.
US_ITERA	This DLL is a clone of the Zemax "Standard" surface, except it uses "dumb" iteration to find the intercept point rather than a closed form expression. The point of this example is to show how to find the surface intercept point if only the sag expression is known. Most polynomial aspheres require this type of iteration because the ray-surface intercept formulas cannot be determined in closed form.
US_LUNE	Models a perfect Luneberg lens GRIN surface.

DLL name	Description

US_MEMS	Models a Micro-Electromechanical System (MEMS), such as a Digital Mirror Device (DMD). The MEMS consists of a 2D array of small rectangular mirrors. The mirrors may tip at any of three angles, each rotated about an axis to point the mirror in any direction. The mirrors may be set to any of the three states by addressing rows, columns, or by individual mirrors if desired to model any state the MEMS can be in. Although the model is strictly geometric, it effectively models where rays are reflected by such a device for an arbitrary setting of the individual mirrors. The parameters on the model are:
	 Nx: Number of X direction mirrors
	 Ny: Number of Y direction mirrors
	 Wx: Total width in lens units in X direction
	 Wy: Total width in lens units in Y direction
	 A0, A1, A2: Angle of mirror in state 0, 1, or 2
	 Rotation Angle: Angle of rotation about the Z axis to tip the mirror
	 Extra Data Value 1: If 0, mirrors are addressed by rows, if 1, by columns, if 2, by individual mirrors.
	 Extra Data Value 2: State of rows/columns/mirrors 1 - 15
	 Extra Data Value 3: State of rows/columns/mirrors 16 - 30
	 Extra Data Value n: State of rows/columns/mirrors 1+15*(n-2) - 15*(n-1)...
	 The rotation angle effectively rotates the plane of tip of the mirror around the local Z axis; with the initial tip plane being around the local x axis. The rotation angle is measured clockwise from the +y axis. The mirror angles are then tipped about the rotated tip direction.
	 The Extra Data Values (EDV) are used to define the state of the rows/columns/ mirrors using a base 3 integer value. To determine the values for any logic state of the MEMS, construct a table similar to the one below, which shows the values for 3 rows or columns or mirrors:
	 `r/c/m: 3 2 1 EDV`
	 `       0 0 0 0`
	 `       0 0 1 1`
	 `       0 0 2 2`
	 `       0 1 0 3`
	 `       0 1 1 4`
	 `       0 1 2 5`
	 `       0 2 0 6`
	 `etc..`
	
	 Note the EDVA is given by:
	 EDV = M1*(3^0) + M2*(3^1) + M3*(3^2) + ...
	 where M1 is the logic state (0, 1, or 2) of the first row/column/mirror and M2 is the logic state of the second row/column/mirror, etc. Up to 15 row/column/mirror values are defined by each EDV.

US_MULTI_ZONE_AS PHERE	This surface models a series of annular even aspheric zones. Each zone has a radius, conic, even asphere coefficients to order 12, and a maximum radial aperture for the zone. Up to 20 concentric zones are supported. Each zone is shifted so the surface is continuous across the zone boundaries.
US_OFFST	The offset surface. This surface simulates an additional propagation of distance that is hidden within the surface. The propagation distance is independent for each wavelength. This allows modeling of systems that have independent focusing of each color channel, such as a digital projector.

DLL name	Description
US_OGIVE	Ogive surface shape. An ogive is identical to the Standard surface, except the axis of rotation of the surface is offset by an amount ro. The surface sag is given by $$z = \frac{cr_g^2}{1 + \sqrt{1 - (1 + k)c^2 r_g^2}}, \text{ where}$$ $$r_g = r_o + \sqrt{x^2 + y^2}.$$
US_POLARIZATION	This DLL shows how to define the electric field or coating data.
US_STAND	This DLL is a clone of the Zemax "Standard" surface type. This is the simplest DLL surface and is a good place to start the study of UDS DLLs.
US_STAND2	This DLL is a clone of the Zemax "Standard" surface type using the UserDefinedSurface2 function and structure definition.

Variable Line Space Grating

The variable line space grating surface models a special grating with straight, but unevenly spaced lines. The grating is described in M. Hettrick and S. Bowyer, "Variable line-space gratings: new designs for use in grazing incidence spectrometers", Applied Optics, Vol. 22, No. 24, p3921 (1983). The grating is designed to form images upon a concave spherical surface. A converging beam is used to illuminate the grating in reflection. The grating lines are parallel to the local x axis, and lie in the x-z plane. Only plane gratings are supported, and the feature has only been tested when the grating is used in reflection (the glass type must be "MIRROR").

Although the hologram surface type could be used to model the ideal grating, in practice holographic construction is not always possible. A compromise is made in the fabrication, where the grating lines are straight and parallel (as opposed to curved for the hologram case). This introduces aberrations into the beam, which this surface model accounts for.

Five parameters are used to describe the surface: M, the diffraction order upon playback; L, the (positive) radius of the concave focal surface, $\cos(\alpha_0)$, the cosine of the angle the incident beam makes with the plane of the grating, $\cos(\beta_0)$, the cosine of the angle the reflected beam makes with the plane of the grating, and λ_0, the construction wavelength. This is the wavelength at which the grating will reflect the axis ray along the angle β_0.

The grating frequency in lines per micrometer is given by

$$T = \frac{1}{\lambda_0} \left[\frac{L\cos\beta_0 - y}{\sqrt{y^2 + L^2 - 2Ly\cos\beta_0}} - \frac{L\cos\alpha_0 - y}{\sqrt{y^2 + L^2 - 2Ly\cos\alpha_0}} \right],$$

where y varies across the grating surface. The variable line space grating surface uses five of the parameter surfaces, but none of the extra data surfaces.

PARAMETER DEFINITIONS FOR VARIABLE LINE SPACE GRATING SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
M	L	$\cos\alpha_0$	$\cos\beta_0$	λ_0

Zernike Fringe Phase

The Zernike Fringe Phase surface has a substrate shape identical to the Standard surface (which supports planes, spheres, and conics) plus additional phase terms defined by the Zernike Fringe coefficients. The surface sag is identical to the Standard surface formula. The additional phase terms deviate and add optical path to the rays as they cross the surface. This surface model is well suited to modeling system aberrations for which measured interferometer data is available. The Zernike Fringe phase surface can also be used to model some holograms and binary optics surfaces. The phase of the surface is given by

$$\Phi = M \sum_{i=1}^{N} 2\pi A_i Z_i(\rho, \varphi)$$

Where N is the number of Zernike coefficients in the series, A_i is the coefficient on the ith Zernike Fringe polynomial, ρ is the normalized radial ray coordinate, φ is the angular ray coordinate, and M is the diffraction order. The Zernike Fringe polynomials are defined in the table given in "Zernike Fringe Coefficients". Zemax supports up to 37 terms. The coefficients A_i all have units of waves. One wave is 2π radians. Note that the phase of the surface is independent of the wavelength; Zemax accounts for the deviation in the ray path based upon the wavelength being traced. If the "Extrapolate" flag is set to 0, the Zernike terms are ignored outside of the normalization radius. If the "Extrapolate" flag is set to 1, then the Zernike terms are considered no matter where the ray lands on the surface; even if the ray lands beyond the normalization radius.

Note that the Zernike Fringe Phase surface describes phase variations (or wavefront error), not surface deformations directly. If you have Zernike coefficient data in terms of surface deformations, as might have been measured with a profilometer, see "Zernike Fringe Sag". See also "Zernike Standard Phase".

PARAMETER DEFINITIONS FOR ZERNIKE FRINGE PHASE SURFACES

Parameter 0	Parameter 1
Diffraction order	Extrapolate

EXTRA DATA DEFINITIONS FOR ZERNIKE FRINGE PHASE SURFACES

Parameter Number	Description
13	Number of terms.
14	Normalization radius. Coordinates are normalized by this value.
15 - 51	Coefficients on Zernike polynomials 1 - 37, respectively, in units of waves.
52 and above	Not used.

The "Number of terms" is used to specify the maximum Zernike polynomial term to be used in calculating the phase. This number is provided to speed the ray tracing calculation; terms beyond this number are ignored.

Zernike polynomials are orthogonal over the unit circle, and so the normalization radius should be set to the radius over which the coefficient data was normalized. Zernike polynomials tend to diverge quite rapidly beyond the normalization radius, and so care should be taken that rays do not strike the surface beyond this radius. Although the ray tracing algorithm may work, the data may be inaccurate. The extrapolate flag may be set to zero to ignore the Zernike terms for rays that land outside the normalization radius.

Zernike Fringe Phase coefficients sign conventions

See "Binary Optic 1" for a discussion of sign conventions.

Zernike Fringe Sag

The Zernike Fringe Sag surface is defined by the same polynomial as the Even Aspheric surface (which supports planes, spheres, conics, and polynomial aspheres) plus additional aspheric terms defined by the Zernike Fringe coefficients. The surface sag is of the form:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=1}^{8} \alpha_i r^{2i} + \sum_{i=1}^{N} A_i Z_i(\rho, \varphi)$$

where N is the number of Zernike coefficients in the series, A_i is the coefficient on the ith Zernike Fringe polynomial, r is the radial ray coordinate in lens units, ρ is the normalized radial ray coordinate, and φ is the angular ray coordinate. The Zernike terms may be decentered from the conic and aspherical terms using parameters 9 and 10. The Zernike Fringe polynomials are defined in the table "Zernike Fringe Coefficients". Zemax supports up to 37 Zernike Fringe terms. The coefficients A_i all have units which are the same as the lens units, such as millimeters or inches. The coefficients α_i have units, and are as defined in the section "Even Asphere". If the "Extrapolate" flag is set to 0, the Zernike terms are ignored outside of the normalization radius. If the "Extrapolate" flag is set to 1, then the Zernike terms are considered no matter where the ray lands on the surface; even if the ray lands beyond the normalization radius.

Note that the Zernike Fringe Sag surface describes surface deformations, not wavefront error directly. If you have Zernike coefficient data in terms of waves of OPD, as might have been measured with an interferometer, see "Zernike Fringe Phase". Also see "Zernike Standard Sag".

PARAMETER DEFINITIONS FOR ZERNIKE FRINGE SAG SURFACES

Parameter 0	Parameters 1-8	Parameter 9	Parameter 10
Extrapolate	$\alpha_1 - \alpha_8$	Zernike decenter X in lens units	Zernike decenter Y in lens units

EXTRA DATA DEFINITIONS FOR ZERNIKE FRINGE SAG SURFACES

Parameter Number	Description
13	Number of terms.
14	Normalization radius. Coordinates are normalized by this value.
15 - 51	Coefficients on Zernike polynomials 1 - 37, respectively, in lens units.
52 and above	Not used.

The "Number of terms" is used to specify the maximum Zernike polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation; terms beyond this number are ignored.

Zernike polynomials are orthogonal over the unit circle, and so the normalization radius should be set to the radius over which the coefficient data was normalized. Zernike polynomials tend to diverge quite rapidly beyond the normalization radius, and so care should be taken that rays do not strike the surface beyond this radius. Although the ray tracing algorithm may work, the data may be inaccurate. The extrapolate flag may be set to zero to ignore the Zernike terms for rays that land outside the normalization radius.

Zernike Standard Phase

The Zernike Standard phase surface has a substrate shape identical to the Standard surface (which supports planes, spheres, and conics) plus additional phase terms defined by the Zernike Standard coefficients. The surface sag is identical to the Standard surface formula. The additional phase terms deviate and add optical path to the rays as they cross the surface. This surface model is well suited to modeling system aberrations for which measured interferometer data is available. The Zernike Standard Phase surface can also be used to model some holograms and binary optics surfaces. The phase of the surface is given by

$$\Phi = M \sum_{i=1}^{N} 2\pi A_i Z_i(\rho, \varphi)$$

where N is the number of Zernike coefficients in the series, A_i is the coefficient on the ith Zernike Standard polynomial, ρ is the normalized radial ray coordinate, φ is the angular ray coordinate, and M is the diffraction order. The Zernike Standard polynomials are defined in the table given in "Zernike Standard Coefficients". Zemax supports up to 231 terms. The coefficients A_i all have units of waves. One wave is 2π radians. Note that the phase of the surface is independent of the wavelength; Zemax accounts for the deviation in the ray path based upon the wavelength being traced. If the "Extrapolate" flag is set to 0, the Zernike terms are ignored outside of the normalization radius. If the "Extrapolate" flag is set to 1, then the Zernike terms are considered no matter where the ray lands on the surface; even if the ray lands beyond the normalization radius.

Note that the Zernike Standard Phase surface describes phase variations (or wavefront error), not surface deformations directly. If you have Zernike coefficient data in terms of surface deformations, as might have been measured with a profilometer, see "Zernike Standard Sag".

PARAMETER DEFINITIONS FOR ZERNIKE STANDARD PHASE SURFACES

Parameter 0	Parameter 1
Diffraction order	Extrapolate

EXTRA DATA DEFINITIONS FOR ZERNIKE STANDARD PHASE SURFACES

Parameter Number	Description
13	Number of terms.
14	Normalization radius. Coordinates are normalized by this value.
15 - 245	Coefficients on Zernike polynomials 1 - 231, respectively, in units of waves.
246 and above	Not used.

The "Number of terms" is used to specify the maximum Zernike polynomial term to be used in calculating the phase. This number is provided to speed the ray tracing calculation; terms beyond this number are ignored.

Zernike polynomials are orthogonal over the unit circle, and so the normalization radius should be set to the radius over which the coefficient data was normalized. Zernike polynomials tend to diverge quite rapidly beyond the normalization radius, and so care should be taken that rays do not strike the surface beyond this radius. Although the ray tracing algorithm may work, the data may be inaccurate. The extrapolate flag may be set to zero to ignore the Zernike terms for rays that land outside the normalization radius.

Zernike Standard Phase coefficients sign conventions

See "Binary Optic 1" for a discussion of sign conventions

Zernike Standard Sag

The Zernike Standard Sag surface is defined by the same polynomial as the Even Aspheric surface (which supports planes, spheres, conics, and polynomial aspheres) plus additional aspheric terms defined by the Zernike Standard coefficients. The surface sag is of the form:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=1}^{8} \alpha_i r^{2i} + \sum_{i=1}^{N} A_i Z_i(\rho, \varphi)$$

where N is the number of Zernike coefficients in the series, A_i is the coefficient on the ith Zernike Standard polynomial, r is the radial ray coordinate in lens units, ρ is the normalized radial ray coordinate, and φ is the angular ray coordinate. The Zernike terms may be decentered from the conic and aspherical terms using parameters 9 and 10. The Zernike Standard polynomials are defined in the table given in "Zernike Standard Coefficients". Zemax supports the first 231 Zernike Standard terms. The coefficients A_i all have units which are the same as the lens units, such as millimeters or inches. The coefficients α_i have units, and are as defined in the section on the "Even Asphere" surface type. If the "Extrapolate" flag is set to 0, the Zernike terms are ignored outside of the normalization radius. If the "Extrapolate" flag is set to 1, then the Zernike terms are considered no matter where the ray lands on the surface; even if the ray lands beyond the normalization radius.

Note that the Zernike Standard Sag surface describes surface deformations, not wavefront error directly. If you have Zernike coefficient data in terms of waves of OPD, as might have been measured with an interferometer, use the Zernike Standard Phase surface instead.

PARAMETER DEFINITIONS FOR ZERNIKE STANDARD SAG SURFACES

Parameter 0	Parameters 1-8	Parameter 9	Parameter 10
Extrapolate	$\alpha_1 - \alpha_8$	Zernike decenter X in lens units	Zernike decenter Y in lens units

EXTRA DATA DEFINITIONS FOR ZERNIKE STANDARD SAG SURFACES

Parameter Number	Description
13	Number of terms.
14	Normalization radius. Coordinates are normalized by this value.
15 - 245	Coefficients on Zernike polynomials 1 - 231, respectively, in lens units.

The "Number of terms" is used to specify the maximum Zernike polynomial term to be used in calculating the surface sag. This number is provided to speed the ray tracing calculation; terms beyond this number are ignored.

Zernike polynomials are orthogonal over the unit circle, and so the normalization radius should be set to the radius over which the coefficient data was normalized. Zernike polynomials tend to diverge quite rapidly beyond the normalization radius, and so care should be taken that rays do not strike the surface beyond this radius. Although the ray tracing algorithm may work, the data may be inaccurate. The extrapolate flag may be set to zero to ignore the Zernike terms for rays that land outside the normalization radius.

Zernike Annular Phase

The Zernike Annular Phase surface has a substrate shape identical to the Standard surface (which supports planes, spheres, and conics) plus additional phase terms defined by the Zernike Annular coefficients. The surface sag is identical to the Standard surface formula. The additional phase terms deviate and add optical path to the rays as they cross the surface. This surface model is well suited to modeling system aberrations for which measured interferometer data is available. The Zernike Annular Phase surface can also be used to model some holograms and binary optics surfaces. The phase of the surface is given by

$$\Phi = M \sum_{i=1}^{N} 2\pi A_i Z_i(\rho, \varphi)$$

where N is the number of Zernike coefficients in the series, A_i is the coefficient on the ith Zernike Annular polynomial, ρ is the normalized radial ray coordinate, φ is the angular ray coordinate, and M is the diffraction order. The Zernike Annular polynomials are defined in the table given in "Zernike Annular Coefficients". Zemax supports up to 231 terms. The coefficients A_i all have units of waves. One wave is 2π radians. Note that the phase of the surface is independent of the wavelength; Zemax accounts for the deviation in the ray path based upon the wavelength being traced. If the "Extrapolate" flag is set to 0, the Zernike terms are ignored outside of the annular region defined by the normalization radius and the obscuration ratio. If the "Extrapolate" flag is set to 1, then the Zernike terms are considered outside the normalization radius. Within the obscuration radius, the Zernike terms are always ignored, no matter how the "Extrapolate" flag is set. The Obscuration is a dimensionless number between 0.0 and 1.0, that defines the annular region size relative to the normalization radius. For example, if the normalization radius is 10.0, then an obscuration of 0.3 would define the annular region of the phase terms to be from 3.0 to 10.0 units.

Note that the Zernike Annular Phase surface describes phase variations (or wavefront error), not surface deformations directly.

PARAMETER DEFINITIONS FOR ZERNIKE ANNULAR PHASE SURFACES

Parameter 0	Parameter 1	Parameter 3
Diffraction order	Extrapolate	Obscuration

EXTRA DATA DEFINITIONS FOR ZERNIKE ANNULAR PHASE SURFACES

Parameter Number	Description
13	Number of terms.

14	Normalization radius. Coordinates are normalized by this value.
15 - 245	Coefficients on Zernike polynomials 1 - 231, respectively, in lens units.

The "Number of terms" is used to specify the maximum Zernike polynomial term to be used in calculating the phase. This number is provided to speed the ray tracing calculation; terms beyond this number are ignored.

Zernike polynomials are orthogonal over the annular region defined by the obscuration ratio within the unit circle, and so the normalization radius should be set to the radius over which the coefficient data was normalized. Zernike polynomials tend to diverge quite rapidly outside the annular region defined by the normalization radius and obscuration ratio, and so care should be taken that rays do not strike the surface outside this region. Although the ray tracing algorithm may work, the data may be inaccurate. The extrapolate flag may be set to zero to ignore the Zernike terms for rays that land outside the normalization radius.

Zernike Annular Phase coefficients sign conventions

See "Binary Optic 1" for a discussion of sign conventions

Zone Plate

This surface is used to model a Fresnel Zone Plate (FZP). A FZP consists of a refractive or reflective surface upon which annular zones of varying depth are cut or etched. Typically, the spacing between grooves is large compared to the wavelength, and therefore FZP's are entirely refractive, and not diffractive, devices. The varying depth of the annular zones imparts a slight phase change to different zones of the beam, due to the change in path length through the material.

The FZP is modeled by altering the local depth of the surface. Edge effects, such as rays striking the side of the zone, and diffraction from the edges, are ignored. The FZP is defined by a normalized optical thickness of each zone. Each zone is spaced a fixed radius apart, and up to 240 zones may be defined. Three parameter values are required: the mode, the Δr, and the reference wavelength, λ_0.

If the mode is zero, the steps are not interpolated, and each zone has a constant optical thickness. This results in discontinuous phase steps, like a staircase. If the mode is unity, then the optical thickness is interpolated between adjacent zones. Although the phase will be continuous, the first derivative of the phase will generally be discontinuous at the zone boundaries. The "Δr" is the step in radius between adjacent zones in lens units. The "reference wavelength" is the wavelength in micrometers used to normalize the optical thickness. For example, if the FZP defines the boundary between a material with index 1.55 and air, with index 1.00, and the reference wavelength is 0.45 micrometers, then an optical thickness of 1 wave corresponds to a groove depth of

$$ T = \frac{\lambda_0}{n_2 - n_1} = \frac{0.45\mu}{0.55} = 0.818182\mu . $$

This thickness is the actual groove depth if the optical thickness is given to be 1.0. Zemax uses this normalized optical thickness for convenience. To specify a zone that introduces one-half wave of phase at the reference wavelength, enter an optical thickness of 0.5. At other wavelengths, the phase will be greater or less depending upon the dispersion of the two adjacent medi The optical thickness is defined at n points, separated by Δr: $r_1, r_2, r_3, \ldots r_n$.specified in the first column of data in the extra data file to import (see the Import section of the Surface Properties). The remaining terms are defined in columns 2 through n+1 in the extra data file.

PARAMETER DEFINITIONS FOR ZONE PLATE SURFACES

Parameter 1	Parameter 2	Parameter 3
Mode	Δr	Ref. Wave

EXTRA DATA DEFINITIONS FOR ZONE PLATE SURFACES

Parameter Number	Description
4 – 12	Unused
13	Number of points
14	Optical thickness at Δr
15	Optical thickness at 2 Δr
16	Optical thickness at 3 Δr
n – 13	Optical thickness at n Δr

Non-Sequential Overview

The following sections are an overview of non-sequential ray tracing in Zemax. This reference article is intended to be used with the NSC Editor section of The Setup Tab in-line help file, and The Analyze Tab (Non-Sequential UI Mode) in-line help file.

Most imaging systems are well described by optical surfaces which are sequential, which means that rays always trace from the object surface to surface 1, then 2, then 3, etc... in a strict sequence. Each ray "hits" each surface once and only once in this predetermined sequence. Rays in sequential mode never trace from surface 3 to 9 then back to 1, for example. The sequential model is simple, numerically fast, and extremely useful and complete for many important cases.

However, there are times when a non-sequential trace is required. Non-sequential means the rays trace in the actual physical order they hit various objects or surfaces, and not necessarily in the order the objects are listed in the software user interface.

NSC Introduction

Non-sequential ray tracing means the rays trace in the actual physical order they hit various objects or surfaces, and not necessarily in the order the objects are listed in the software user interface. Note rays in a non-sequential trace may hit the same object repeatedly, and entirely miss other objects. Generally, the order in which objects are hit by rays depends upon the object geometry and the angle and position of the input ray. Objects which require or at least benefit from non-sequential ray tracing include prisms, light pipes, lens arrays, reflectors, and Fresnel lenses. Certain types of analysis, such as stray or scattered light effects, are only practical in a completely non-sequential environment.

Traditionally, lens design programs that supported surfaces (rather than 3D objects) for sequential ray tracing would implement non-sequential ray tracing using the same surface model; the rays would simply intersect surfaces in a possibly out of sequence order.

The disadvantage of using surfaces in a more general non-sequential approach is that surfaces do not adequately describe most optical components. For example, lenses not only have a front and back surface, they also have edges and perhaps flattened outer faces for mounting. Light may intercept, and then refract or reflect from these additional surfaces normally ignored by sequential surface ray tracing codes. Complex prisms, such as a dove or roof prism, contain many faces, and the rays may intersect these faces in a complex order depending upon the input angle and position of the ray.

To support these types of components in a very general and accurate way requires the use of full 3D solid object models instead of just 2D surfaces. Zemax calls this type of ray tracing non-sequential components, or NSC, which is different from non-sequential surfaces or NSS. NSC ray tracing in Zemax supports all of the following features:

Definition and placement of multiple sources, objects and detectors.

Real radiometric and photometric units; including watts, lumens, lux, phot, footcandles, and others.

Automatic determination of ray-object intersection order.

Automatic detection of reflection, refraction, and total internal reflection (TIR).

Support for a very wide range of 3D objects, including diffractive optics.

Polarization ray tracing and arbitrary thin film coatings.

Statistical models for scattering, including Lambertian, Gaussian, and ABg.

Automatic ray splitting for efficient analysis.

For more information on setting up a NSC group, defining objects, and tracing rays through NSC, see the following section called "NSC Ray Tracing".

NSC Ray Tracing

The following sections explain how to perform non-sequential ray tracing in Zemax.

Methods of using NSC ray tracing

Zemax supports two distinct methods of using NSC ray tracing:

Tracing rays through an NSC group that is part of an otherwise sequential system (NSC with ports).

Tracing rays through an NSC group that contains all the objects of interest (NSC without ports).

Although the method of defining and placing objects within a NSC group is the same for both methods, the details of how rays are launched, what analysis may be performed, how energy distributions are determined, and what types of systems are best modeled by each method, are considerably different. NSC with and NSC without ports are described below.

NSC With Ports

This is tracing rays through an NSC group that is part of an otherwise sequential system.

When non-sequential object or objects are part of an otherwise sequential system, NSC with ports is generally the best approach to use. An example of this system would be a point or extended surface object represented by rays which follow a sequential path through one or more conventional lenses, then follow a non-sequential path through a prism or light pipe before illuminating the image surface.

This method requires the use of ports for rays to enter and leave the NSC group. Ports are described in detail in the section "NSC ray tracing with ports". When using ports, rays are launched from defined field positions on the object surface, and these rays ignore source and detector objects within the NSC group. Rays must leave the NSC group via the exit port; then continue through the remainder of the sequential system.

When using analysis features such as ray fans, spot diagrams, and MTF, only rays which enter and leave the NSC group through ports are considered. All the usual sequential Zemax system data, such as field positions and pupil sizes, determine the properties of rays entering the NSC group. All normal Zemax analysis, such as ray fans and spot diagrams, are still available (although the data may be meaningless depending upon the properties of the NSC system).

NSC Without Ports

This is tracing rays through an NSC group that contains all the objects of interest.

Some systems have no sequential paths or portions at all; such as headlamp reflectors, complex light pipes, or general illumination systems. It is also possible to analyze ghost, stray, and scattered light properties of nominally sequential systems (such as a camera lens or telescope) by placing the entire system in a non-sequential group and performing non-sequential ray tracing on the entire model. These types of systems and analysis are better suited to NSC without ports.

This method requires the use of NSC sources to launch rays. When not using ports, rays are launched only from defined NSC sources, and these rays consider detector objects within the NSC group.

Since rays launched from sources within the NSC group cannot leave through the exit port; the only available analysis is the ray distribution and energy as determined using detector objects.

Although the method of defining and placing objects within a NSC group is the same for with and without ports, the details of how rays are launched, what analysis may be performed, how energy distributions are determined, and what types of systems are best modeled by each method, are considerably different. NSC with and NSC without ports are described below.

NSC With and Without Ports

The 3D Layout, Wireframe, Solid Model, and Shaded Model analysis features are all capable of simultaneously displaying BOTH rays coming from the sequential entry port as well as any sources defined for use in NSC without ports. Rays entering from the entry port do not interact with any NSC detectors; and rays launched from NSC sources do not interact with the entry port, the exit port, or any optics defined outside the NSC group.

Therefore it is advised, though not required, that when using NSC without ports (i.e. you have defined sources among your NSC objects) that you set the number of rays in these layout features to zero. This will make for a less confusing display. There are times when showing the sequential rays coming from the entry port and non-sequential rays on the same plot is useful, especially when placing baffles in a stray light analysis.

Alternatively, there is a 3D Layout feature which is specific to NSC, which shows only objects and rays defined within a single NSC group.

Paraxial data and ray tracing with NSC

There is no paraxial ray tracing in NSC. When paraxial rays are traced to a non-sequential surface, the equivalent real rays are traced instead. Nearly all paraxial data, such as focal length and F/#, are meaningless for non-sequential systems. Some NSC systems are well behaved in the sense that real rays close to the axis behave much as idealized paraxial rays would. However, NSC systems with obscurations or objects only partially in the path of the beam will not be able to correctly predict any paraxial data, and paraxial data should be considered meaningless for these systems.

How to use NSC With Ports

Ray tracing through a group of NSC objects using ports is accomplished with the following basic steps:

1) A Non-Sequential Components surface is inserted in the Lens Data Editor. This surface becomes the entry port for the non-sequential group.

2) The Non-Sequential Components surface parameters are used to define the location of the exit port for the non-sequential group.

3) Objects are defined in a list associated with the Non-Sequential Components surface.

4) Zemax traces a ray sequentially to the entry port, then non-sequentially within the NSC group, until the ray strikes the exit port.

5) Rays entering the NSC group through the entry port cannot split.

The Non-Sequential Components surface has parameters which determine where the rays will exit the NSC group as described below.

The Entry Port

The Non-Sequential Components surface acts like a plane, sphere, or conic aspheric surface whose location is determined by the previous surfaces in the Lens Data Editor, in the usual way. The surface shape may be hyperhemispheric to allow acceptance of rays over a full 4π steradians. The Non-Sequential Components surface is the entry port into a group of objects which will be traced non-sequentially. The entry port is how rays get into the NSC group. No objects should be placed as to touch or surround the entrance port.

The Exit Port

There are 9 parameters used in the definition of the Non-Sequential Components surface:

Draw Ports? If 0, no ports are drawn, if 1, draw entry, 2, draw exit, 3 draw both.

Exit Location X The x position of the exit port relative to the entry port.

Exit Location Y The y position of the exit port relative to the entry port.

Exit Location Z The z position of the exit port relative to the entry port.

Exit Tilt About X The rotation about the local X axis of the exit port.

Exit Tilt About Y The rotation about the local Y axis of the exit port.

Exit Tilt About Z The rotation about the local Z axis of the exit port.

Order If the order flag is zero, then the above locations and tilts are done in the following order: decenter x, decenter y, decenter z, rotate around global z, rotate around global y, rotate around global x. If the flag is any value other than zero, then the order is reversed. This follows the same convention as the sequential coordinate break surface when using order flag = 0.

Reverse Rays If this flag is 0, then Zemax assumes the non-sequential group acts like a refractive lens. If this flag is 1, then Zemax assumes the non-sequential group acts like a mirror. For example, if rays entering the non-sequential group travel in a positive direction with respect to the local z axis, and leave the exit port still traveling in the positive local z direction, the flag should be zero. If the rays reverse direction relative to the incoming direction, then the reverse rays flag should be 1.

Exit Port Diameter This is in lens units. The diameter of the circular exit port is defined by the *semi-diameter of the surface following* the Non-Sequential Components surface. Note any additional aperture may be placed on the exit surface if another aperture shape is required.

These parameters define the location and size of the exit port relative to the entry port. If the exit port is located exactly at the entry port, then rays will immediately exit the non-sequential group without striking any objects. This generally means the exit location z parameter must not be zero.

The glass column of the Non-Sequential Components surface is also used to define the "background" material and index of refraction of the media in which NSC objects are placed. The surface after the Non-Sequential Components surface acts like a Standard plane surface oriented in the coordinate system after the decenters and tilts have been applied.

Note the exit port position is the same as the surface following the Non-Sequential Components surface, and its location in 3D space is determined by the parameters of the Non-Sequential Components surface. The thickness of the Non-Sequential Components surface is not used; only the location and tilt parameter values. The exit port should not be placed at the same location as, or within the glue distance (see "Glue Distance In Lens Units") of the entrance port; otherwise rays entering the NSC group will immediately strike the exit port and not intersect any of the objects within. No objects should be placed as to touch or surround the exit port.

Ray Tracing

Getting Rays In

A ray leaves the object surface, and traces through the lens in the usual sequential fashion until it reaches the Non-Sequential Components surface. The ray is then sent into the group of components associated with that surface, and the non-sequential tracing begins.

Once inside the NSC group, 3 things can happen to a ray:

1) It can hit the exit port.

2) It can hit nothing at all.

3) It can hit one of the objects within the group.

If the ray hits the exit port, the ray coordinates and the direction cosines are computed on the exit port, and the ray then traces sequentially again through the remaining surfaces of the lens.

If the ray hits nothing at all, then the ray tracing is terminated, and the ray trace function returns a "ray missed" error on the following surface (since the ray never struck the exit port, which is always the next surface in the sequential portion of the ray trace).

If the ray strikes an object in the NSC group, then the ray will either reflect, refract, total internal reflect (TIR), or be absorbed, depending upon the properties of the object struck. Rays entering the NSC group through the entry port cannot split. If the ray is absorbed, the ray trace is terminated and a ray miss error is returned, otherwise the new ray coordinates and direction cosines are computed, and the process repeats until one of the following conditions is met:

1) The ray hits the exit port.

2) The ray hits no object.

3) The ray is absorbed.

4) The ray has intercepted more than the maximum allowed number of objects (see "Maximum Intersections Per Ray").

Cases 1, 2, and 3 are handled exactly as described above. In case 4, even though the ray technically still can be traced, it is terminated to prevent infinite loops from occurring. In this case, the ray trace returns a ray miss error.

Getting Rays Out

When a ray strikes the exit port, the coordinates and direction cosines of the ray in the coordinate system of the exit port are computed, and then the ray traces sequentially through any remaining surfaces. If one of the following surfaces is another Non-Sequential Components surface, then the process begins again for the components defined for that group. Note rays within one NSC group cannot "see" objects defined in another group, even if they physically share the same location in space; nor can the rays "see" surfaces outside the present NSC group.

How to use NSC Without Ports

Ray tracing through a group of NSC objects without using ports is accomplished with the following basic steps:

1) Change the program mode to Non-Sequential Mode in the Setup tab.

2) Insert sources, objects, and detectors in the Non-Sequential Components editor.

There are no entry or exit ports to consider; these are ignored when using NSC without ports. The only data that needs to be defined outside the NSC Editor are:

The wavelengths to be used for ray tracing (in the System Explorer)

The glass catalogs to be used (in the System Explorer)

The coating definitions (in the coating file associated with the current lens)

Ray Tracing

Getting Rays In

To get rays into the NSC group, define one or more sources. Zemax supports point, rectangular, elliptical, user defined, and other source models. Each source object has the following parameters defined (and perhaps other parameters as well):

 # Layout Rays Defines how many random rays to launch from the source when creating layout plots.

 # Analysis Rays Defines how many random rays to launch from the source when performing analysis.

Power (units) Power is the total power over the defined range of the source. The power units are specified by the system source units. See "Source Units" for details.

Wavenumber The wavenumber to use when tracing random rays. Zero means polychromatic; which chooses ray wavelengths randomly with the weighting defined on the wavelength data dialog box.

Note multiple sources may be superimposed with different powers and wavenumbers to create correct polychromatic sources. Sources may be placed anywhere without restriction (even inside objects). Once the ray is launched, non-sequential tracing begins.

Once inside the NSC group, 2 things can happen to a ray:

1) It can hit nothing at all. If the ray hits nothing at all, then the ray tracing for this ray is terminated.

2) It can hit one of the objects within the group. If the ray strikes an object, then the ray will either reflect, refract, total internal reflect (TIR), scatter, split, diffract, or be absorbed; or a combination of these, depending upon the properties of the object struck.

Detecting Rays

If the ray strikes a detector object; the pixel which the ray struck is determined, and the total pixel energy is incremented by the ray energy. Detectors may be absorbing, reflecting, transmissive, or refractive.

The process repeats until one of the following conditions is met:

1) The ray hits no object.

2) The ray is absorbed.

3) The ray has intercepted more than the maximum allowed number of objects.

4) The total number of ray segments exceeds the maximum allowed number.

5) The relative or absolute energy of the ray falls below the minimum threshold.

The limitations on number of objects, ray segments, and ray energy are defined in the "Non-Sequential" section of the System Explorer. Cases 1 and 2 are handled exactly as described above. In case 3, 4, and 5, even though the ray technically still can be traced, it is terminated to prevent infinite loops from occurring. Rays may be launched, and detectors reset, from the Detectors, Ray Trace/Detector Control.

Launching rays for analysis

Rays may be launched, and detectors reset, from the Detectors, Ray Trace/Detector Control.

Combining NSC with and without ports

The 3D Layout, Wireframe, Solid Model, and Shaded Model analysis features are all capable of simultaneously displaying BOTH rays coming from the sequential entry port as well as any sources defined for use in NSC without ports. Rays entering from the entry port do not interact with any NSC detectors; and rays launched from NSC sources do not interact with the entry port, the exit port, or any optics defined outside the NSC group.

Therefore it is advised, though not required, that when using NSC without ports (i.e. you have defined sources among your NSC objects) that you set the number of rays in these layout features to zero. This will make for a less confusing display. There are times when showing the sequential rays coming from the entry port and non- sequential rays on the same plot is useful, especially when placing baffles in a stray light analysis.

Alternatively, there is a 3D Layout feature which is specific to NSC, which shows only objects and rays defined within a single NSC group.

Object Placement

The NSC Editor allows specification of object placement. This section is repeated in the Setup Tab in-line help section that describes use of the NSC Editor. The conventions and restrictions on placing objects in the NSC group is critically important. Objects may be placed anywhere in 3D space; and objects may be placed with respect to any other object. Objects may also be placed entirely inside of other objects, or may be placed adjacent to other objects.

See Non-Sequential Objects, Non-Sequential Sources, and Non-Sequential Detectors.

Coordinate System

Each object's position is defined by 6 parameters: the x, y, and z coordinates, and the rotation about the x, y, and z axes. Zemax first decenters in x, y, and z (decenters are orthogonal so the order does not matter). Then Zemax tilts about the local x axis (which rotates the y and z axes to new orientations), then tilts about the new y axis (which rotates the x and z axes), then finally tilts about the new z axis. This is the same convention as for the coordinate break surface when using order flag = 0. The conversion from object local coordinates to global coordinates can be written in equation form as

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix},$$

where the g subscript indicates the global coordinate, o is the offset, and l is the local object coordinate. The matrix R is the rotation matrix, which relates the orientation of local and global coordinates. These equations can be written more compactly as G = O + RL, where G is the global coordinate vector, O is the offset vector, R is the rotation matrix, and L is the local coordinate vector. For more information on properties of the rotation matrix, see "Global Coordinate

Reference Surface". The order of rotations may also be reversed, see "Use Global XYZ Rotation Order" under the Type section of the Object Properties.

Reference Objects

It is often useful to reference an object's position and rotation relative to another object. This is useful especially when placing related objects in a group, and then decentering or tilting the entire group.

The object to which the coordinates are relative to is the "reference object". The default reference object is object 0, which is the vertex of the Non-Sequential Component surface. If a positive number greater than zero is specified, then the coordinates of the object are referenced to the location and rotation of the specified object. This is an "absolute" reference object.

If the reference object number is negative, then the reference object is determined by adding the current object number to the negative reference object number. This is a "relative" reference object. For example, if the reference object is -3 on object 8, the reference object will be 5 because 8 - 3 = 5. Relative reference objects are particularly useful when copying and pasting groups of objects; this is easiest if all the objects in the group use relative references to the first object in the group.

When a reference object is used, the rotation and offset matrices then become:

$$G' = O' + R'G$$
$$G' = O' + R'[O + RL]$$
$$G' = [O' + R'O] + [R'R]L$$

Any number of coordinate reference nesting is supported; so that object 9 could be placed in the coordinate frame of object 5 which in turn is placed in the frame of object 3. The only restriction is that the reference object must precede in the object list the object whose coordinates are being referenced. To modify the reference object without the need to manually recompute the object coordinates and angles, see the "Modify Reference Objects".

Placement Inside, Adjacent, Overlapping

Placing sources within other objects, and the use of absolute and relative values for the "Inside Of" column are described in "Placing sources inside objects".

Objects can be combined to make more complex objects, by placing one object inside of, adjacent to, or overlapping another object. What determines the ray tracing properties of such a compound object depends upon the position and type of the various objects and whether or not they touch or overlap. Here the word touch means that one or more points on the boundary surface of one object is in the same location in 3D space as a point on a boundary surface of another object. Mirror objects may be placed anywhere, even in contact with or partially or fully inside any other object without restriction. Rays will always reflect from mirror surfaces back into the medium the rays had been traveling through.

Glue Distance In Lens Units

When two NSC objects are placed in contact, such as a lens touching one face of a prism, numerical roundoff will cause the ray tracing algorithm to sometimes detect a very tiny distance between the two objects. This can also occur when objects are rotated in 3D space and placed close, but not exactly, next to one another because of the finite number of digits entered in the spreadsheet editor.

The glue distance is the distance below which the objects are considered in contact. It is important that objects not be separated by distances very close to the glue distance. If it is intended that two objects be in contact, then the maximum spacing between the objects should be several times smaller than the glue distance. If it is intended that the objects be separated, then the distance between the object should be several times larger than the glue distance. Object spacings very close to the glue distance will yield inconsistent ray tracing or geometry errors. This should be avoided by adjusting either the object spacing or the glue distance.

The glue distance also determines the minimum propagation length for ray tracing. If a ray-object intersection is less than the glue distance away from the previous intercept, the intercept is ignored.

The glue distance is also used in the tolerance associated with ray tracing to general curved surfaces. Zemax will iterate until the error in the ray-surface intercept solution is less than one-fifth of the glue distance.

In the majority of cases, no adjustment should be made to the glue distance parameter. The glue distance must be no smaller than 1.0E-10 and no larger than 1.0E-03.

Nesting object limits

There is a user defined limit on the maximum number of nested objects. This defines an upper limit on how many objects can be placed inside each other. For example, if the maximum number of nested objects is 3, then object 3 may be placed inside of 2 which is placed inside of object 1. There may be any number of groups of objects each nested 3 deep in this case. The limit applies to the total nesting in any one collection of objects, however, there may be any number of such collections within the NSC group. The maximum number of nested objects is set on the Non-Sequential section of the System Explorer. Setting the nesting limit no higher than required for the system being modeled conserves memory usage, although the minimum setting is 4.

Nesting volumes

Volumes of refracting material are more complicated, because Zemax must keep track of the index of refraction through which the ray is propagating. The rule to remember is: if a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface or volume at that point.

> **If a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface or volume at that point.**

For example, if a diffraction grating lens is object #1, and a non- grating lens of the same thickness and radii made of air or glass is placed inside the first lens is object #2; then rays that strike in the zone including both objects #1 and #2 will act as though they just hit object #2.

This allows defining objects with "holes" and other compound objects. Objects may be touching at one or more faces, nested or not, or may partially overlap, to create a wide range of compound solid shapes.

Nesting surfaces

The nesting rules defined above apply to solid volumes of refractive or reflective material. Some special surfaces may also be "nested" in the sense that more than one surface may exist at the same point in space. The rules are similar to those that apply to volumes, but surfaces cannot be refracting.

The same rule applies to surfaces as does for solids: if a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface at that point.

If a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface at that point.

For surfaces, there are a few rules to consider when more than one surface exists at the ray intercept point:

1) The last surface listed will determine the properties of the surface.

2) If the last surface listed is a mirror, the ray will reflect.

3) If the last surface listed is an absorber, the ray will be absorbed.

4) If the last surface listed is neither a mirror nor an absorber, the ray will ignore the surface.

5) Surface objects may not share boundaries with volume objects unless the surface object is reflective or absorbing, or unless the volume object is listed after the surface object; in which case the volume defines the properties of the common boundary.

6) Only standard surface objects may share boundaries, support for nesting of other surface object types may be added in future versions of the software.

Refraction and reflection from NSC objects

All NSC objects allow specification of a material on the NSC Editor (gradient index media are defined separately, see "Defining GRIN media for non-sequential ray tracing"). The reflective, refractive, and absorptive properties of an object depend upon the object type, whether or not the object describes a surface or a volume, and the name of the material specified. The material name may be blank, "MIRROR", "ABSORB", a glass name, a table glass name, a MIL number designation, or a model glass.

No material defined
Blank materials are assumed to indicate an index of unity, which may or may not be the background index of the non-sequential space (which is set by the index of the NSC surface on the Lens Data Editor).

MIRROR or ABSORB
A material of MIRROR indicates the object reflects all incident rays. A material of ABSORB indicates the object absorbs all incident rays. Note that a material of ABSORB absorbs all incident rays even if a reflective coating is applied to the face of the NSC object.

Material defined by a glass name
Homogeneous glasses are defined in the glass catalogs. For more information, see "Specifying which glass catalogs to use".

Material defined by a table glass
To define a table glass, see "Using table glasses".

Material defined by MIL number
A 6 digit MIL number, such as 517642, may be used to define an approximate material model. For more information on MIL number glasses, see "Using MIL number glasses".

Material defined by a glass model
The glass model uses 3 independent terms to define a typical dispersion curve. The model glass is more flexible and accurate than the MIL glass, but not as good as actual measured data. For more information see "Using model glasses".

Object properties by material
The properties of objects are described by the following table.

Object Type	Material is blank or glass	Material is "MIRROR"	Material is "ABSORB"
Volume	All faces are refractive.	All faces are reflective.	All faces are absorbing.
Surface	Object is ignored.		
Polygon Object (volume)	If a face in the POB file is marked as reflective or absorbing, then the face remains reflective or absorbing. Refractive faces remain refractive.	If a face in the POB file is marked as absorbing, then the face remains absorbing. All other faces become reflective.	If a face in the POB file is marked as reflective, then the face remains reflective. All other faces become absorbing.
Polygon Object (surface)	Object is ignored.	If a face in the POB file is marked as absorbing, then the face remains absorbing. All other faces become reflective.	If a face in the POB file is marked as reflective, then the face remains reflective. All other faces become absorbing.

Reflective objects are usually surfaces rather than volumes. Hollow light pipes can also be modeled as a group of flat reflectors. Some reflectors do enclose a volume; the interior of which no ray will ever see. Examples include rectangular

and spherical volumes with all surfaces set to be reflective. Any volume with a material name of "MIRROR" will be all reflective.

Refractors must always enclose a volume. Volume refractors include lenses, prisms, and solid light pipes. Absorbers may be surfaces or volumes with the material name set to "ABSORB". Polygon Objects are composed of faces, and each individual faces may be refractive, reflective, or absorptive.

Diffraction from NSC objects

Some NSC objects have one or more diffractive faces; such as the Diffraction Grating, Binary 1, and Binary 2 objects. These objects refract or reflect rays as well as diffract them, according to the grating period or phase and the diffraction order and wavelength. For any ray, if the diffraction order being traced does not satisfy the grating equation, then the energy of that ray will refract or reflect along the zero order path.

All diffractive NSC objects support a parameter to defined the "order" for diffraction from the object. This order is called the "primary" diffraction order. Sequential rays which enter the non-sequential group through the entry port will diffract only along the primary order.

Rays which originate from a non-sequential source will diffract only along the primary order if ray splitting is off. If ray splitting is on, then the diffraction may be optionally controlled by the settings on the "Diffraction" tab of the Object Properties dialog box. This tab includes optional settings which will split the ray by order; allowing more than one diffraction order to be simultaneously traced.

For more information on diffraction, see the "Diffraction" section of the "Object Properties".

Polarization and thin film coatings

Rays traced through non-sequential components may be done while accounting for polarization effects, or polarization may be ignored. The initial polarization state for a ray is determined by the source properties, see "Sources" section of "Object Properties". If polarization ray tracing is being used, transmission, reflection, and absorption of optical energy is accounted for at all surfaces. Bulk absorption is also accounted for. Thin film coatings significantly affect transmission and reflection properties of optical surfaces. Surfaces are initially uncoated, but coatings may be applied to surfaces or group of surfaces.

Coatings on surfaces in contact

If two surfaces are in contact, such as two 45-45-90 prisms placed so that one face of each prism is in contact with the other, then a coating may be applied "between" the surfaces in contact.

This is accomplished using the same convention described above in the section "Object Placement". The rule is: the LAST object listed in the NSC Editor determines the properties of the interface between two objects.

For example, to place a thin metal coating on the interface between two prisms arranged to form a beam splitter, the first prism object listed should have the contact surface "uncoated", while the second object listed should have the contact surface coated with the appropriate thin film coating. Rays striking this interface from either side will see the correct

coating, and the ray transmission and/or reflection will be correctly computed. Note prisms modeled as POB objects can have different coatings applied to different faces, so some faces may be anti-reflection coated while others are coated with a reflective coating.

Scattering

Scattering may occur at any ray-surface intersection. Scattering properties are defined for each face on an object. A face is a collection of one or more surfaces on an object upon which the same optical properties are applied. For example, for a singlet lens, there are 3 faces: the front face, the back face, and all remaining faces (which include the edges and squared faces around the edges). The default scatter model is "No scattering", which means no scattering will occur. The resulting ray is called the unscattered or the "specular" ray (even if the surface is not actually reflective).

The settings described in the following sections are found in the Coat/Scatter and Scatter To sections of the NSC Object Properties.

Fraction to scatter and number of scatter rays

If a scattering model other than "No scattering", "ABg", "ABg File" or "IS Scatter Catalog" is selected, the "Fraction to Scatter" must be defined. This fraction must be between 0.0 (no rays will be scattered) and 1.0 (every ray will be scattered). For the ABg and ABg File scattering models, the fraction to scatter is determined by the ABg parameters, see "Defining ABg data". For the IS Scatter Catalog scattering model, the fraction is always 1, with no user control over this input, as the total integrated scatter for this model is included explicitly in the measured data.

If ray splitting is off, the decision to scatter or not to scatter is made by the generation of a single random number between 0.0 and 1.0. If this random number is larger than the fraction to scatter, the ray will not scatter, otherwise, the ray will scatter. For example, if the fraction to scatter is 1.0, the ray always scatters. If the fraction to scatter is 0.0, the ray will never scatter. If the fraction to scatter is 0.25, then on average one out of four rays will scatter. All of the energy of the ray follows the randomly generated scatter path. The number of scatter rays has no affect if ray splitting is off.

If ray splitting is on, then Zemax will split the specular ray into one or more scattered rays, while still possibly tracing the specular ray. The specular ray will receive a fraction of the original energy equal to (1.0 - f) where f is the fraction to scatter. The remaining energy will be divided equally among the one or more scattered rays. The number of scatter rays determines how many scatter rays will be generated. For example, if the fraction to scatter is 1.0, then the specular ray will receive zero energy and will no longer be traced; and all the energy will be divided equally among the scattered rays. If the fraction to scatter is 0.0, no scattered rays will be traced, and the specular ray retains all the original energy. If the fraction to scatter is 0.25 and the number of scatter rays is 5, then the specular ray will receive a relative energy of 0.75, and each of the 5 scattered rays will have a relative energy of 0.05. If the number of scatter rays is set to zero, then the fraction to scatter is ignored and no scattering occurs. The maximum number of scatter rays is 300, although much smaller values are typically used.

Scatter models

Scatter models are defined in terms of a probability distribution function. When Zemax scatters a ray, a new direction of propagation is chosen. The direction is chosen using a probability function and one or more random numbers. The net effect is that if many rays are traced, the resulting scattered ray distribution would approach the probability distribution function. The following figure defines the vectors used to describe the scattering models.
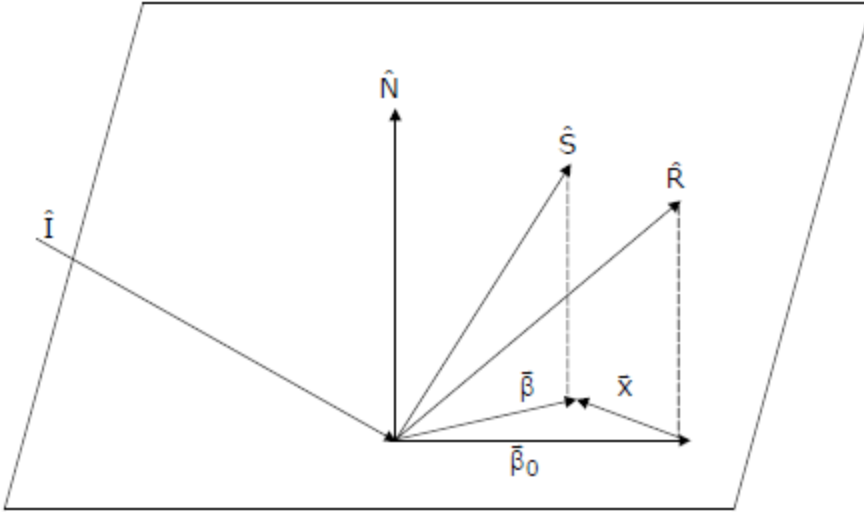
The normal vector N defines the orientation of the surface at the ray-surface interception point. The incident ray vector is I , the specular ray vector is R , and the scattered ray vector is S. The specular vector may be the reflected or the refracted ray vector; the figure shows this ray as the reflected ray vector for simplicity. Note N, I, R, and S are all unit vectors. The projection of the specular and scattered ray vectors down to the surface are denoted by β_0 and β, respectively. The projections are not unit vectors; the vector β_0 has a magnitude equal to $\sin\theta_r$, while β has a magnitude of $\sin\theta_s$, where θ_r and θ_S denote the angle between the normal vector and the specular and scattered ray angles, respectively. The vector (β -β_0) is denoted x, and if |x| tends to zero, the scattered and specular vector become the same. How x is determined depends upon the scatter model chosen.

The maximum magnitude for the vectors β_0 and β is 1.0. However, because these vectors do not generally point in the same direction the maximum possible magnitude for the vector x is 2.0. The vector x must also result in a vector β that lies within the unit circle of the projected plane.

Bi-Directional Scatter Distribution Function

The Bi-Directional Scatter Distribution Function (BSDF) is defined as the scattered radiance per unit incident irradiance, or

$$BSDF(\theta_i, \phi_i, \theta_s, \phi_s) = \frac{dL_s(\theta_s, \phi_s)}{dE_i(\theta_i, \phi_i)}$$

where θ is measured from the normal, and ϕ is the azimuthal angle, and the subscripts i and s refer to incident and scattered directions, respectively. Note BSDF has units of inverse steradians. The BSDF may also be defined as a function of the vector x rather than the polar coordinates θ and ϕ . In general, the BSDF is a function of the incident angle and wavelength. The term BSDF can refer to two separate functions, the BRDF and BTDF, for reflective and transmitted distributions, respectively.

The integral of the BSDF over all possible scatter angles (a hemisphere) is called the Total Integrated Scatter (TIS) and is defined by

$$TIS(\theta_i, \phi_i) = \int\int BSDF(\theta_i, \phi_i, \theta, \phi) \cos\theta \sin\theta \, d\theta \, d\phi$$

where the i indicates the incident direction. For all scatter models other than ABg, the BSDF is normalized to yield a TIS equal to the "fraction to scatter" parameter. For the ABg model, the TIS must be less than 1.0, and the TIS indicates the total fraction of energy that scatters. All remaining energy is assumed to be specular.

Available scatter models

There are seven scattering models available: none, Lambertian, Gaussian, ABg, ABg File, BSDF, and user defined. Each available scatter model is defined in the following sections.

No scattering

No modification to the specular ray is made, the BSDF is zero and the magnitude of the vector $\vec{x}$ is zero.

Lambertian scattering

For Lambertian scatter, the scattered ray projection vector $\vec{\beta}$ has equal probability anywhere in the unit circle, and the BSDF is just $1/\pi$. The scattered intensity is proportional to $\cos\theta_s$. Note Lambertian scattering is independent of the ray incident angle. Most diffuse surfaces are nearly Lambertian.

Gaussian scattering

Gaussian scattering is of the form:

$$BSDF(\vec{x}) = Ae^{\frac{-|\vec{x}|^2}{\sigma^2}}$$

where A is a normalizing constant. The resulting distribution is rotationally symmetric in direction cosine space, no matter what angle the specular ray makes with respect to the surface normal. The dimensionless value σ determines the width of the Gaussian distribution on the projected plane. Values of σ greater than about 5.0 yield a BSDF that is nearly Lambertian. For this reason, the maximum allowed value of σ is 5.0.

ABg model scattering

The ABg scattering model is a widely used method for defining the BSDF. This scattering model is generally a good model to use when the scattering is mainly due to random isotropic surface roughness, and the scale of the roughness is small compared to wavelength of light being scattered. These assumptions are generally valid for polished optical surfaces. The ABg BSDF is given by

$$BSDF(\vec{x}) = \frac{A}{B + |\vec{x}|^g}$$

The following restrictions are placed on the coefficients of this model: A must be greater than or equal to 0.0, and B must be greater than 1E-12 unless g = 0.0. If g is 0.0, B may be 0.0. If A is 0.0, no scattering will occur. If g is 0.0 (note g may be any value, positive or negative, but it typically is between 0.0 and 3.0), then the BSDF is constant:

$$BSDF(\vec{x}) = \frac{A}{B + 1}$$

and the resulting scattering is effectively Lambertian. If A, B, and g are set to provide a relatively flat BSDF curve, it is far faster numerically to use the Lambertian model instead.

A typical ABg BSDF curve is shown below for the parameters A = 0.002, B = 0.0005, g = 2.0.
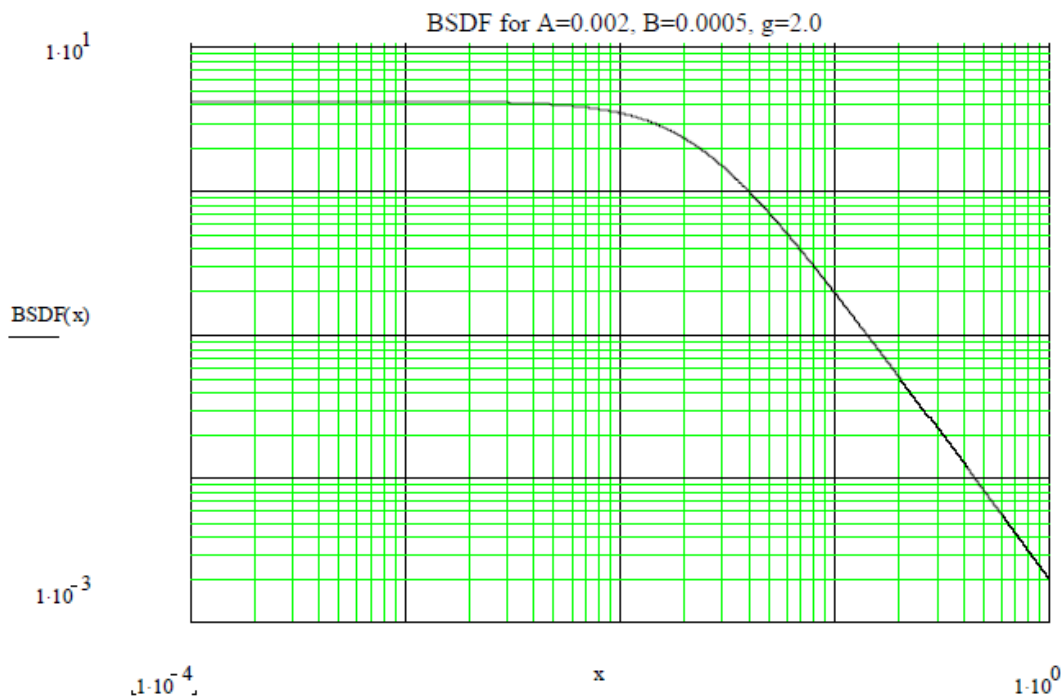
The ABg BSDF model has several interesting properties (when g is not zero):

When $|\vec{x}|^g \ll B$, the BSDF curve flattens out to a value of A/B as $|\vec{x}|$ goes to zero

When $|\vec{x}|^g \gg B$, the BSDF becomes a straight line on the log-log plot, with a slope of -g.

The transition point between the flat and sloped parts of the curve occurs at $|\vec{x}| = \beta_t$, where $\beta_t = 10^{\frac{\log B}{g}}$

Defining ABg data

It is frequently the case where the same ABg data will be applied to many objects in the optical system. To avoid the need to type in the same data redundantly, and to simplify the task of editing the ABg data, Zemax provides an ABg data catalog. The catalog may be edited directly in Zemax; see "ABg Scatter Data Catalogs". Note the ABg data defined must integrate over all possible scatter angles to a total value of less than 1.0 to conserve energy.

Zemax allows definition of separate ABg data for reflection and refraction. If the specular ray reflects or refracts, the ray is subsequently scattered using the appropriate coefficients. The ABg data names which appear in the dialog box control are those defined by the ABg scatter data file. The ABg data file to be used is selected from the "Files" section of the System Explorer.

Zemax first decides whether or not to scatter the ray, as described in the section "Scattering" above. If the ray scatters, a scattered ray is randomly generated. The ray direction is generated so that if a large number of rays were scattered, the appropriate BSDF function would result.

For important information on using scatter models with a source defined by a Source Color spectrum, see "Wavelength scaling of ABg data".

ABg File scattering

The ABg File scattering model allows a sum of ABg profiles to be used to define the scattering properties of a surface. The profiles to use are specified in a text file. The text file must have a .ABGF extension and must be located in the <data>\ABg_Data folder (see "Folders"). All of the profiles specified in the ABGF file must be defined in the currently loaded ABg Data File (see "ABg Data File"), and all profiles must be specified using upper case letters (regardless of capitalization of the profile name in the ABg Data File).
The syntax of the ABGF file is:

```
Number of profiles Weights_Are_Absolute<optional> Profile_1_Name
Profile_1_Weight
Profile_2_Name Profile_2_Weight
...
```

The first line of the file should contain either 1 or 2 numbers: the first defining the number of ABg profiles contained in the ABGF file, and the second (optional value) defining a flag that indicates whether the weights provided for each profile are relative or absolute. If this flag is omitted or set to zero, the profile weights are taken to be relative; otherwise the profile weights are taken to be absolute. In either case, the BSDF values for each profile are multiplied by the corresponding profile weight and then added together. If the profile weights are taken to be relative, the sum of BSDF values is divided by the sum of the weights, and the resulting scatter distribution corresponds to a weighted sum of the defined profiles. Conversely, if the profile weights are taken to be absolute, the sum of the BSDF values is not normalized, and the resulting scattering distribution corresponds to an absolute sum of the profiles. In this latter case, it is up to the user to confirm that the TIS resulting from an absolute sum of the specified ABg profiles does not exceed unity for any wavelength or any angle of incidence.

The remaining lines of the file simply contain the name of each ABg profile to sum and the corresponding weight. The name of the profile should be entered exactly as it appears in the ABg Scatter Data Catalog (see "ABg Scatter Data Catalogs"). Each of the profiles is treated as it would be under the ABg scatter model (see "ABg model scattering"). Thus, ABg data for each profile will be scaled with wavelength if a reference wavelength is defined in the ABg Scatter Data Catalog for that profile, and the ABg data for each profile will be interpolated between the two points with angles of incidence closest to that of the incident angle for the scattered ray. The wavelength scaling and interpolation over angle of incidence will be performed for each profile specified in the ABGF file before all profiles are summed.

The maximum number of profiles which may be specified in an ABGF file is 20. Please contact Zemax technical support if you require this limit to be increased.

BSDF scattering

The BSDF scattering model allows the use of tabular BSDF data for defining the scattering properties of a surface. Data are provided via text files. Files must follow the BSDF Data Interchange file format described in the article entitled "BSDF Data Interchange File Format Specification", which may be found in the Zemax Knowledge Base (http://zemax.com/support/knowledgebase).

Zemax allows definition of separate BSDF data for reflection and refraction. If the specular ray reflects or refracts, the ray is subsequently scattered using data from the appropriate input file. For a file to be available for use in this scatter

model, it must have a .BSDF extension (as indicated in the article describing the file format), and be located in the <data>\Scatterdata folder (see "Folders"). The maximum number of characters that the file name can contain (including the .BSDF extension) is 60.

A full description of the model and its use is provided in the Knowledge Base article entitled "How to Use Tabular Data to Define the Surface Scattering Distribution" also available on the Zemax web site.

IS Scatter Catalog scattering

The IS Scatter Catalog scatter model allows the use of data measured with the Radiant Zemax IS-SATM device for defining the scattering properties of a surface. Data are provided via binary ISX files, a library of which is provided with Zemax (see "Download IS® Scatter Catalog Data").

Zemax allows definition of separate data for reflection and refraction. If the specular ray reflects or refracts, the ray is subsequently scattered using data from the appropriate input file. For a file to be available for use in this scatter model, it must have a .ISX extension and be located in the <data>\Scatterdata folder (see "Folders"). The maximum number of characters that the file name can contain (including the .ISX extension) is 60.

This model is functionally very similar to the BSDF scattering modeling described above, and as such many of the descriptions provided in the Knowledge Base article entitled "How to Use Tabular Data to Define the Surface Scattering Distribution" apply to this model as well. However, Zemax can also read spectral information from the ISX file, if such information exists. Thus, the variation of scattering with wavelength is supported in this model.

The user is allowed to specify the angular resolution with which data from the ISX file is read. Current Zemax options are 5 degrees, 2 degrees, and 1 degree, in which the value refers to the angular resolution in both the polar (radial) and azimuthal directions. Please contact Zemax technical support if you require a different resolution option and/or higher resolution.

The IS-SA may be used to measure either the front or back side of a scattering sample, or both sides. The user must then specify which side of the sample to use in defining the scattering distribution for the surface. If the user selects a side for which data are not available, Zemax will issue an error message and force selection of the other available side.

The ISX file contains the full energy distribution for light scattering from the sample (in reflection and/or refraction). As such, no degrees of freedom exist to specify an independent face property, coating or scatter fraction for any surface on which this scatter model is applied, and those controls will be greyed out when the IS Scatter Catalog scatter model is selected for a surface.

To model scattering in reflection and transmission off of a thin window, the "Thin Window Scattering" option should be selected for all faces of the window (see "Thin Window Scattering").

User defined scattering

Completely general surface scattering may be defined via an external program called a Dynamic Link Library (DLL). Sample DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs".

Defining an object to use DLL defined surface scattering

To make an object use a DLL defined scatter function, choose the "User Defined" scattering type from the object properties dialog box, then select from the available DLL scatter functions listed in the "DLL Name" box.

DLL parameters

Each DLL may use up to 6 user defined data values as parameters in the computation of the scattering properties. These values are defined by the DLL and are only used by the DLL.

Creating a new DLL

The DLL must include two functions:
    UserScatterDefinition
    UserParamNames

When a ray is scattered, Zemax passes to the UserScatterDefinition function the local x, y, and z coordinates on the surface, the local normal vector, the local specular ray vector, polarization data, and other parameter data as defined by the user. UserScatterDefinition then is required to determine the following values:

    Whether or not the ray actually scatters
    The direction cosines of the scattered ray The attenuation of the ray, if any
    Optionally the new electric field vector (if this is not provided Zemax will make a reasonable guess)
    Optionally the BSDF and TIS if Importance Sampling is used (see "Importance Sampling").

These values are returned to Zemax and are used to continue the trace. The function UserParamNames is used to define the names of all used parameters. These names appear in the Coating/Scattering tab of the object properties dialog box. Surface scattering DLLs must be placed in the <program>\DLL\SurfaceScatter folder. See "Folders".

Note that the "Fraction to scatter" control is active when User Defined scatter is selected. Only the fraction of energy which is allowed to scatter is sent to the DLL for possible scattering. However, the DLL may choose to not scatter the ray (see for example the sample DLL in the next section).

More details can be found in the source code of the sample surface scattering DLL files provided with Zemax.

Sample surface scattering: TwoGaussian.DLL

The surface scattering sample TwoGaussian.DLL supports a scatter model which is the superposition of 4 scattering models: specular, Lambertian, and two individual Gaussian models. The DLL uses the parameter data to define the fraction of energy which scatters according to the Gaussian Sigma 1, the fraction which scatters according to Sigma 2, and the fraction that is Lambertian. The remaining fraction, if any, is specular. To see how this is accomplished in the DLL, review the source code file Two Gaussian.c provided with Zemax.

Sample surface scattering: K-correlation.DLL

The surface scattering sample K-correlation.DLL supports a scatter model which follows the K-correlation distribution, as described in the paper "K-correlation power spectral density & surface scatter model" by Michael G. Dittman (Proc. of SPIE Vol. 6291, 62910R, 2006). The inputs to the DLL include the surface roughness (in micrometers), the characteristic surface wavelength (multiplied by 2π), and the logarithmic slope of the BSDF. More details are provided in the article entitled "How to Model Surface Scattering via the K-correlation Distribution" in the Zemax Knowledge Base. The source code K-correlation.c is provided with Zemax as a sample file.

Sample surface scattering: RI_BSDF.DLL

The sample RI_BSDF.DLL supports a model in which tabular data are used to define the scattering properties of a surface. This DLL is considered obsolete. Any users wishing to define scattering via tabular input data should use the built-in BSDF scatter model (see "BSDF scattering").

Thin Window Scattering

When this option is selected, rays will only scatter in reflection upon entering a scattering volume (i.e. when entering the window from air) and will only scatter in transmission when exiting the volume (i.e. when exiting the window into air); otherwise rays will follow the specular ray path. This option is generally used when modeling a thin scattering volume in which the scatter data were measured, i.e. in which the BSDF and/or IS Scatter Catalog scatter models are selected. When measuring scattering in reflection or transmission from a thin volume, the reflected (BRDF) and transmitted (BTDF) measurements encompass all of the scatter paths in the volume. Since each single set of data (BRDF or BTDF) encompass all real scatter paths, all of these paths need to be consolidated when modeling scatter in Zemax. Thus, with selection of the Thin Window Scattering option, scattering only happens once in reflection and in transmission from the volume.

How to model scattering efficiently

A very large number of rays may need to be traced to find a relatively small number of scattered rays that strike an object of interest, such as a detector. There are two ways to improve the efficiency of the scattering analysis.

The first method is to scatter a ray according to the scatter distribution, but only trace the ray if the ray propagates towards an object of interest. This method may be implemented by defining a "Scatter To" list of objects. The Scatter To method works well for wide angle scatter (such as Lambertian scatter) and when the object of interest subtends a relatively large angle as seen from the scattering surface.

The second method is to always scatter the ray towards the object of interest, and then to normalize the energy the ray carries to account for the probability the ray would have actually scattered in that direction. This method is called "Importance Sampling". Importance Sampling is generally superior to the Scatter To method if the scatter is narrow angle or the object of interest subtends a relatively small angle as seen from the scattering surface.

The Scatter To list

The Scatter To feature speeds up scattering analysis by ignoring scattered rays which do not propagate directly towards an object of interest. The Scatter To list is very similar to the Consider Objects list (see "Defining paths using the Consider Objects list"). The Scatter To list is a string of integer object numbers separated by spaces. A scattered ray will only be traced if the ray intersects one of the objects listed or the object that scattered the ray. For example, suppose object 2 defines the Scatter To list as "3 4". If a ray scatters off object 2, it will only be traced if the ray will intersect objects 2, 3, or 4. Note that any other objects are ignored, so scattered rays will miss any object not explicitly listed (or the scattering object itself), even if that object lies directly in the path between the scatter point and the listed scatter to object.

Choosing an object on the Scatter To list does not guarantee that a scattered ray will be traced to that object. The scattered rays are generated based upon the scattering properties of the object the incident ray strikes (see the "Scatter To" section of the Object Properties). Zemax will generate the scattered rays, then ignore those rays that do not intersect any of the listed objects. This method does not ensure that a ray will scatter toward a listed object. However, the number of scattered rays (see "Fraction to scatter and number of scatter rays") may be made large so that it is more probable some of them will head in the desired directions.

To turn off the Scatter To feature, leave the list blank. In this case, all scattered rays will be traced. The key difference between the Scatter To list and the Consider Object list is that the Scatter To list only applies to scattered rays, while the Consider Object list applies to both specular and scattered rays.

Importance Sampling

Importance Sampling speeds up scattering analysis by always choosing scattered rays that will propagate in a desired direction. To determine the direction for the scattered ray, a target sphere is defined. The target sphere is defined by position, size, and limit. The target sphere position is defined by the selection of an object number. The target sphere will then be placed at the origin of the selected object. Any object type may be used, including Null objects. The size parameter defines the radius in lens units of the target sphere. The limit parameter is used to define the maximum solid angle in Steradians of the target sphere as seen from the scatter point. If the solid angle of the target sphere as seen from the scatter point exceeds the limit value, Importance Sampling will not be used for that scattered ray. The purpose of this feature will be discussed below.

When the ray scatters from the surface, the direction and subtended solid angle of the target sphere as seen from the scatter point are computed. A ray direction is randomly selected from within this solid angle cone with uniform probability. This will be the scattered ray direction. Note that the direction is not dependent upon the BSDF of the surface.

Because the ray direction is determined without consideration of the BSDF, the energy the ray carries must be modified to get correct ray tracing results. The amount of energy the scattered ray will carry depends upon the BSDF and TIS of the surface as follows:

$$E' = E \frac{BSDF}{TIS} \Omega \cos \theta$$

where E is the energy allocated to the scattered ray before importance sampling is applied, E' is the amount of energy in the ray traced towards the target sphere, Ω is the solid angle of the target sphere as seen from the scatter point, θ is the angle between the local normal and the scattered ray, and BSDF and TIS are the usual scattering function and total integrated scatter (see "Bi-Directional Scatter Distribution Function"). Note this energy renormalization is only accurate if the BSDF is reasonably uniform over the solid angle of the target sphere. If Ω becomes large compared to the variation of the BSDF, the energy normalization will not be accurate. For this reason, a limit value may be defined. If the target sphere is close enough to the scatter point so that Ω exceeds the limit, then Importance Sampling is not used for that scattered ray.

Up to 6 independent target spheres may be defined for each object. If one scatter ray is selected, the first target sphere is used. If two scatter rays are selected, then the first two target spheres are used, and so on up to 6 scattered rays. If more than 6 scattered rays are selected, the seventh and subsequent rays scatter according to the BSDF and Importance Sampling is not used. Energy normalization is complex and somewhat prone to statistical errors when the number of scattered rays is not equal to the number of target spheres. For the fastest, most accurate results, use a number of scattered rays that is exactly equal to the number of target spheres.

Bulk scattering

The object bulk scattering method is set in the "Bulk Scatter" section of the Object Properties window. Bulk (volumetric) scattering models random scattering of rays while propagating through a solid object. Zemax supports 3 modes for the bulk scattering of a solid object: No Bulk Scattering, Angle Scattering, and DLL Defined Scattering.

No Bulk Scattering

If no bulk scattering is selected, rays will propagate through the solid without scattering.

Angle Scattering

Angle scattering uses a simple model for scattering within a solid. Rays traveling a distance x within the media have an integrated probability of having been scattered given by

$$p(x) = 1.0 - e^{-\mu x}, \text{ where } \mu = \frac{1}{M},$$

and the symbol M is the mean free path in lens units. Note that as x increases, the probability that the ray has scattered asymptotically approaches 1.0. Setting this expression to a randomly selected value between 0.0 and 1.0, then solving for x, yields a randomly generated path length with the correct statistics. If this path length is greater than the distance the ray propagates to the next ray-object intersection, then no scattering occurs; otherwise, the ray scatters at the specified position along the propagation direction.

Once the position is determined, the scattering is modeled by a new ray direction being chosen. For angle scattering, a random angle is chosen so that the new ray direction lies uniformly distributed within a cone that makes an angle with respect to the current ray direction. The semi-angle of the cone is one of the parameters to the model, and this parameter should be set to between 0.0 and 180 degrees. The latter value will scatter the ray randomly in any direction. Once the ray scatters, Zemax automatically will adjust the polarization vector and randomize the phase of the scattered ray.

DLL Defined Scattering

If the angle scattering model is not sufficient, more complex bulk scattering functions may be defined via an external program called a Dynamic Link Library (DLL). Sample DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs".

Sample bulk scattering: Poly_bulk_scat.DLL

The sample bulk scattering file poly_bulk_scat defines the scatter probability as follows:

$$P(\theta) = \sum_{i=0} c_i \theta^i$$

where θ is the polar angle of the scattered ray with respect to the unscattered ray direction. The source code file poly_bulk_scat.c is provided with Zemax as a sample file.

Sample bulk scattering: Henyey-Greenstein-bulk.DLL

The Henyey-Greenstein bulk scatter model defines the scatter probability as follows:

$$P(\theta) = \frac{1 - g^2}{4\pi(1 + g^2 - 2g\cos\theta)^{3/2}}$$

where θ is the polar angle of the scattered ray with respect to the unscattered ray direction. The parameter g can be any value from -1 to 0.9999. If the input value for g is less than -1, g is set to -1. If the input value for g is greater than

0.9999, then g is set to 0.9999. If the input value for |g| is less than 1.0e-4, then g is set to 1.0e-4. The source code file Henyey-Greenstein-bulk.c is provided with Zemax as a sample file.

Sample bulk scattering: Rayleigh.DLL

The Rayleigh bulk scattering model defines the scatter probability according to Rayleigh theory for an unpolarized input beam:

$$P(\theta) = (3(1 + (cos\theta)^2))/(8\lambda^4)$$

where θ is the polar angle of the scattered ray with respect to the unscattered ray direction and λ is the wavelength. The inputs to the DLL are the Reference Wavelength (i.e. the wavelength associated with the input value for the Mean Path) and the Transmission (which allows for energy to be lost during each scattering event). More details are provided in the article entitled "Bulk Scattering with the Rayleigh Model" in the Zemax Knowledge Base. The source code file Rayleigh.c is provided with Zemax as a sample file.

Sample bulk scattering: Mie.DLL

The Mie bulk scattering model defines the scatter probability according to Mie theory. The algorithm used in Zemax was taken from the book "Absorption and Scattering of Light by Small Particles" by Craig F. Bohren and Donald R. Huffman (John Wiley & Sons, 1983). The inputs to the DLL include the particle index, the particle size (in micrometers), and the particle density (in cm-3). More details are provided in the article entitled "How to Simulate Atmospheric Scattering using a Mie model" in the Zemax Knowledge Base. The source code file is not distributed with Zemax.

Sample bulk scattering: Phosphor.DLL

The Phosphor bulk scattering model is meant to represent a phosphor material in which both fluoresce and scattering can occur. Light in a particular wavelength range (defined as "blue") enters the medium in which the DLL is applied, and that light can either be converted to another wavelength (i.e. undergo fluorescence) or simply bulk scatter. When the input light undergoes fluorescence, the resultant light is emitted into a 180 degree cone angle with equal probability (i.e. the input light "scatters" according to the Angle Scattering model). When the input light scatters with fluorescing, it does so according to the Mie distribution. The fluoresced light (or any other input light not in the "blue" wavelength range) may also scatter according to the Mie distribution. The inputs to the DLL include the mean-free path for fluorescence, the wavelength range defining the "blue" wavelength range, and the particle parameters associated with the Mie distribution. More details are provided in the article entitled "Modeling a White Light Source using a Phosphor" in the Zemax Knowledge Base. The source code file is not distributed with Zemax.

Defining an object to use DLL defined bulk scattering

To make an object use a DLL defined bulk scatter function, choose the Bulk Scattering tab from the object properties dialog box, then select Model: "DLL Defined Scattering". The available DLL functions are then listed in the "DLL:" control.

DLL parameters

Each DLL may use between zero and 16 user defined data values as parameters in the computation of the media properties. These values are defined by the DLL and are only used by the DLL.

Creating a new DLL

The DLL must include two functions:
    UserBulkDefinition
    UserParamNames

While ray tracing through a solid using DLL defined bulk scattering, Zemax passes to the UserBulkDefinition function the current propagation length along the ray, various ray data values, and other parameter data as defined by the user. UserBulkDefinition then is required to determine if the ray will scatter at some point along the propagation length. If the ray will scatter, then UserBulkDefinition must determine the following values:

    The position at which the ray will scatter
    The new direction cosines of the ray
    The attenuation of the ray, if any
    Optionally the new electric field vector (if this is not provided Zemax will make a reasonable guess)

These values are returned to Zemax and are used to continue the trace. The function UserParamNames is used to define the names of all used parameters. These names appear in the Bulk Scatter tab of the object properties dialog box. Bulk scattering DLLs must be placed in the <program>\DLL\BulkScatter folder. See "Folders".

Coherence Length Modeling

By default, non-sequential sources in Zemax are either purely monochromatic, or simulated to be polychro- matic by tracing more than one monochromatic wavelength. Physically, no source is perfectly monochromatic. A more realistic model for a nominally monochromatic source is to account for the range of wavelengths emitted by the source. A convenient parameter is the coherence length; defined as

$$\Delta x = \frac{c}{\Delta f}$$

where Δx is the coherence length, c is the speed of light in vacuum and Δf is the range of frequencies emitted by the source, also called the bandwidth. A more convenient representation is the range of wavelengths emitted defined in terms of the coherence length:

$$\Delta \lambda = \frac{\lambda^2}{\Delta x}$$

Using this expression, it is easy to compute the variation of optical path length (phase) as the ray propagates.

This is modeled in Zemax by randomly choosing a wavelength within $\pm(\Delta \lambda)/2$ for purposes of computing the optical path and thus the phase. Zemax chooses the random wavelength with a uniform distribution. The greater the path difference between two rays from the same source, the greater the difference in phase between the rays will be; which means coherent effects such as fringe visibility will be reduced.

Zemax actually traces the ray using the central wavelength. The randomized wavelength is only used to compute the effects of coherence length on the phase. Therefore, only coherent computations will exhibit the phase degradation. To

turn off coherence degradation, either set the coherence length to a very large number or to zero. To make sources completely incoherent, set the coherence length to a small but non-zero number.

Defining GRIN media for non-sequential ray tracing

Any solid enclosed volume may be made of a homogeneous material (the default assumption) or the media may be of a gradient index. All gradient index materials used by non-sequential objects are defined in a separate program called a Dynamic Link Library (DLL). Numerous DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs".

Defining an object to be of gradient index material

To make an object media by of a gradient index material, choose the GRIN tab from the object properties dialog box, then select "Use DLL defined GRIN media". The available gradient index materials are then listed in the "DLL:" control.

Discussion on maximum step size for GRIN objects

The maximum step size determines the tradeoff between ray tracing speed and accuracy. The exact value required depends upon the rate at which the index of refraction changes and the desired accuracy of the computations. If the maximum step size is too large, the ray trace results will have large errors, and some rays will miss objects they would otherwise hit, or vice-a-versa.

GRIN DLL parameters

Each DLL may use between zero and 12 user defined data values as parameters in the computation of the media properties. These values are defined by the DLL and are only used by the DLL.

Creating a new GRIN DLL

The DLL must include two functions:
    UserGrinDefinition
    UserParamNames

While ray tracing through a GRIN media, Zemax passes to the UserGrinDefinition function the current x, y, and z location of the ray (in the objects local coordinate system), the wavelength, and other parameter data as defined by the user. UserGrinDefinition then is required to compute the following four values:
    n (the index of refraction)
    n * dn/dx (the index times the derivative with respect to the x direction)
    n * dn/dy (the index times the derivative with respect to the y direction)
    n * dn/dz (the index times the derivative with respect to the z direction)

These four values are returned to Zemax and are used to trace the ray through the grin media.

The function UserParamNames is used to define the names of all used parameters. These names appear in the GRIN tab of the object properties dialog box. GRIN DLLs must be placed in the <program>\DLL\GradientIndex folder. See "Folders".

CAD Objects and the Grid_Gradient DLL

The Grid_Gradient DLL is based on a three dimensional array of index data, and it is essential that the object to which this GRIN type is applied shares the same coordinate system. This is particularly important for CAD objects; it is highly recommended that the origin of the CAD object be on the front face. It should also be noted that the accuracy of the CAD representation is the limiting factor in the accuracy of the ray tracing; native Zemax objects are recommended where possible.

Sample GRIN DLLs

The following sample GRIN DLLs are provided with Zemax.

DLL name	Description
GRIN1	Defines a GRIN media of the form: $$n = N_0 + N_{r1}r + N_{r2}r^2,$$ similar to the sequential Gradient1 surface type. This sample DLL also supports array dx and array dy values to specify a grin variation which is periodic over the specified intervals. This option is used for arrays of gradient index lenslets. If both the array dx and array dy values are zero, the gradient is not periodic.
GRIN2	Defines a GRIN media of the form: $$n^2 = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{r6}r^6 + n_{r8}r^8 + n_{r10}r^{10} + n_{r12}r^{12},$$ similar to the sequential Gradient2 surface type.
GRIN3	Defines a GRIN media of the form: $$n = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{r6}r^6 + n_{z1}z + n_{z2}z^2 + n_{z3}z^3,$$ similar to the sequential Gradient3 surface type.
GRIN4	Defines a GRIN media of the form: $$n = n_0 + n_{x1}x + n_{x2}x^2 + n_{y1}y + n_{y2}y^2 + n_{z1}z + n_{z2}z^2,$$ similar to the sequential Gradient4 surface type.

GRIN5	Defines a GRIN media of the form: $$n = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{z1}z + n_{z2}z^2 + n_{z3}z^3 + n_{z4}z^4 \, ,$$ similar, but not identical to, the sequential Gradient5 surface type. Note there is no dispersion in this grin model (no dependence on wavelength).
GRIN6	Defines a GRIN media of the form: $$n = n_0 + n_1 r^2 + n_2 r^4 \text{, where}$$ $$n_x = A_x + B_x \lambda^2 + \frac{C_x}{\lambda^2} + \frac{D_x}{\lambda^4} \text{, for all three values of } n_x \, ,$$ which is similar to the sequential Gradient6 surface type.
GRIN9	Defines a GRIN media of the form: $$n = n_0 \left[1.0 - \frac{A}{2} r^g \right] \text{, where}$$ where both A and n_0 are functions of wavelength: $$A(\lambda) = \left[K_0 + \frac{K_1}{\lambda^2} + \frac{K_2}{\lambda^4} \right]^2 \text{, and}$$ $$n_0 = B + \frac{C}{\lambda^2}.$$ This is similar to the sequential Gradient9 surface type, with the additional parameter "g" which is fixed at 2.0 for the sequential surface Gradient9.
GRIN10	Defines a GRIN media of the form: $$n = n_0 + n_{y1}y_a + n_{y2}y_a^2 + n_{y3}y_a^3 + n_{y4}y_a^4 + n_{y5}y_a^5 + n_{y6}y_a^6 \, ,$$ where $y_a = |y|$ and the $||$ symbols indicate the absolute value. This form of gradient has a discontinuity at the plane $y = 0$, and the gradient is bisymmetric about the $y = 0$ plane. This is similar to the sequential Gradient10 surface type, but without the dispersion model.
GRADIUM	Defines a GRIN media of the form: $$n = \sum^{11} n_i \left(\frac{z + \Delta z}{z_{max}} \right)^i$$
Grid_Gradient	Defines a GRIN media based on data provided in a text file, similar to the sequential Grid Gradient surface type. See "Grid Gradient".

TORUS	Defines a GRIN media of the form:
	$$n = N_0 + N_{r2} t^2 \,,$$
	where t is the distance from an axis of rotation defining the toroidal

Defining DLLs for ray splitting at diffractive surfaces

Any diffractive surface may split rays that started at a non-sequential source, by order, to allow for simultaneous tracing of multiple diffracted orders. How many rays split, and how much energy is in each order, may be defined externally in a separate program called a Dynamic Link Library (DLL). Numerous DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs".

Defining an object to use the diffraction DLL

To make a diffractive object use the DLL, choose the Diffraction tab from the object properties dialog box, then select "Split by DLL function". The available DLLs are then listed in the "DLL:" control.

Diffraction DLL parameters

Each DLL may use between zero and 12 user defined data values as parameters in the computation of the diffractive surface properties. These values are defined by the DLL and are only used by the DLL.

Creating a new Diffraction DLL

The DLL must include two functions:
    UserDiffraction
    UserParamNames

While ray tracing through a diffractive surface, Zemax passes to the UserDiffraction function the current x, y, and z location of the ray (in the objects local coordinate system), the wavelength, the present order being traced, and other ray specific data and parameter data as defined by the user. UserDiffraction then is required to compute the relative energy in the specified order, and return this value to Zemax. Zemax will refract or reflect the ray into the specified diffraction order and give the resulting ray the specified fraction of the input ray energy computed by the DLL. If the new child ray has sufficient energy, the ray is traced.

The Diffraction DLL format also supports as an option for the DLL to compute ALL output properties of the ray; including direction cosines, electric field polarization vector, and transmitted energy. If the DLL also computes these properties, the DLL must set a flag indicating the values were passed back to Zemax for further ray tracing. See the sample DLLs for documentation on this feature.

The function UserParamNames is used to define the names of all used parameters. These names appear in the Diffract tab of the object properties dialog box. See the sample DLLs for the use of this feature. Diffraction DLLs must be placed in the <program>\DLL\Difractive folder. See "Folders".

Ray splitting

Generally, when a ray strikes a surface, part of the energy will be reflected, part will be transmitted, and depending upon the surface properties, part may be absorbed. Ray splitting refers to the ability of Zemax to compute both the reflected and refracted paths, and then continue to trace both rays. Zemax also supports an option to randomly choose either the reflected or the refracted path rather than both, see "Simple Ray Splitting". Rays reflected from otherwise refractive interfaces are commonly called ghost reflections.

Once a ray splits, each of the "child" rays will in general strike another object, and the rays may split again and again. After many ray-object intersections, the total number of rays can become extremely large, so controls must be placed on the ray tracing to ensure that the computation will eventually end.

There are several ways to restrict the amount of ray splitting that occurs:

Maximum number of ray-object intersections: This control defines how many times a ray along any path, from the original source parent ray to the final ray-object intersection, may intersect another object.

Maximum number of ray segments: A ray segment is that portion of a ray path from one intersection to the next. When a ray is launched from the source, it travels to the first object. That is 1 segment. If the ray then splits into 2 rays, each of those are another segment (for a total of 3). If each of those rays split again, there will be 7 segments. Generally, the number of ray segments grows far faster, and needs to be set much larger, than the number of ray-object intersections does.

Minimum relative ray intensity: As each ray splits, the energy decreases. The relative ray intensity is a lower limit on how much energy the ray can carry and still be traced. This parameter is a fraction, such as 0.001, relative to the starting ray intensity from the source. Once a child ray falls below this relative energy, the ray is terminated.

Minimum absolute ray intensity: This parameter is very similar to the minimum relative ray intensity, except it is absolute in system source units rather than relative to the starting intensity. If this is zero, the absolute ray intensity threshold is ignored. The initial intensity of each ray is always given by the source intensity divided by the total number of analysis rays for that source. The number of layout rays is not used to determine the initial intensity of rays, even for those rays drawn on layout plots. Source units are defined on the Units tab of the System, General dialog box. See "Source Units" for details.

All of these settings are defined on the Non-Sequential section of the System Explorer. See the System Explorer section of The Setup Tab in-line help file for details.

Ray splitting and polarization

Because accurate reflection and transmission computation requires polarization information, ray splitting is only allowed when performing polarization ray tracing.

Ray splitting can be turned off, and in this case the transmitted path is always taken at a refractive interface unless the ray TIRs; the reflected path is of course always taken if the surface is a mirror.

The following figure shows some of the ray paths possible when rays are split. There is only 1 input ray drawn.
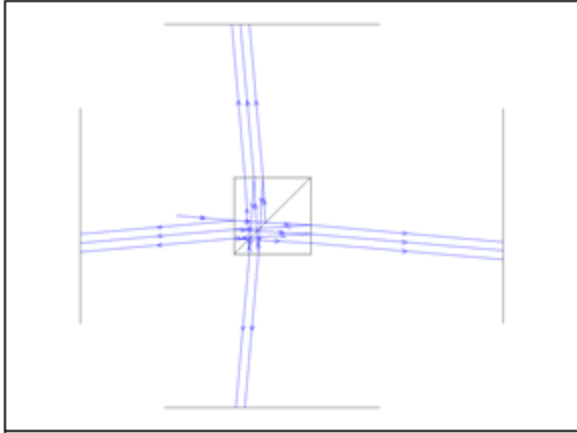
When using ray splitting, the number of rays being traced can become enormous if the minimum relative ray energy is set too low. For example, for rays traced inside a cube of glass with 50% coatings on all surfaces, a relative ray intensity of 0.01 will yield about 256 ray segments. If the relative ray intensity is set to 1E-8, about 270 million ray segments will be generated for EACH ray traced into the cube! Systems with relatively low reflectivity (such as AR coated optics) will not generate as many ray segments because the intensity of the reflected paths falls so much more quickly. Still, it is advised to set the relative ray transmission reasonably high, around 0.001, until the model is working well and more detailed results are needed.

Because the total number of rays gets so large, the 3D Layout plots can become very cluttered. One way to reduce the amount of rays drawn is to use the "filter" string to suppress drawing of some of the rays, see "The filter string".

Putting it all together

The real benefit of the non-sequential ray tracing, polarization, scattering, and ray splitting features occurs when they are all used together to determine ray distributions and detected energies. The order in which these effects are all accounted for is as follows:
1. The object interface the ray strikes is computed first.
2. If the surface is reflective or the ray TIRs, the reflected path is chosen.

Otherwise,
1. If ray splitting is on, the reflected and the refracted path are both determined.
2. If ray splitting is off, the refracted path is chosen.
3. The ray(s) are scattered if scattering is enabled.
4. The ray(s) intensity and polarization vectors are adjusted.
5. The ray(s) continue on to the next object interface.

On 3D Layouts and other analysis features which draw NSC objects and rays, the effects of sources, ray splitting, and scattering may be seen from the randomly generated rays on these plots. Both ray splitting and ray scattering may be selected on or off on these plots to isolate their effects. The layout plots are extremely useful for qualitatively seeing the effects of sources, objects, scattering, and ghost or other reflections.

To accurately compute quantitative energy distributions on the placed detectors generally requires a larger number of rays than the qualitative analysis of the layout plots. This is why the number of layout and number of analysis rays are defined separately in the source parameter listing.

The Ray Trace Control

The Ray Trace Control is used to launch and trace rays for analysis using defined sources, objects, and detectors. The controls on the dialog box are described below.

For more information, see the "Ray Trace" tool in the Analyze Tab in-help file.

Ray database (ZRD) files

Ray data may be saved for subsequent analysis by checking on "Save Rays" and providing a file name with a ZRD extension in the edit box provided. As the rays are traced, the ray data will be saved to the file specified. Saving the ray data may slow down the ray tracing significantly, because of the (relatively) slow speed of the hard disk access. However, it is often convenient to trace the rays once; then do multiple analysis on the resulting data. Rays contained in a ZRD database may be drawn on layout plots rather than tracing new rays; see NSC 3D Layout for details.

Zemax supports three different ZRD file formats, each of which offers various combinations of file size, compression, and data content. The ZRD file formats are binary to keep the file as compact as possible.

All ZRD formats use the same file extension "ZRD". The three letter format names are only used to distinguish the formats in the documentation, and are not used as file extensions.

These are the three different ZRD file formats available:

Uncompressed Full Data (UFD) - All data from the ray trace in an uncompressed format. This is the format to use if you have an external program that reads or writes ZRD files. UFD format makes the largest files, and is also the slowest to read and write. For a detailed description of the UFD format see "The ZRD Uncompressed Full Data Format (UFD)".

Compressed Basic Data (CBD) - This format stores a subset of the full data, including only status, level, hit object, parent, xybin, lmbin, x, y, z, and intensity. The ray coordinates and intensity are stored only to floating point (32 bit) rather than the usual double (64 bit) precision to save disk space. Although direction cosine data is not explicitly stored in the file, the cosines are automatically recreated from the x, y, and z coordinate data during the file read. CBD files are dynamically compressed in a way that makes the file unsuitable for read/write operations outside of Zemax. CBD results in the smallest file size and fastest writing. Generally a file in CBD format will be about 10% as large as the UFD format. Note that data missing from this format is not available for subsequent analysis when reading the ZRD file.

Compressed Full Data (CFD) - All data from the ray trace in a compressed format. All double precision numbers are converted to floating point precision to save space, but in all other respects this format contains all the data from the ray trace, including polarization, index, phase and other data. This format is about 20% as large as the UFD format if polarization is not used, and about 35% as large if polarization is used.

ZRD Format And Version Numbers

ZRD files contain a version number and format number. Zemax will check these numbers when reading the file. Older version files are automatically converted by Zemax into the current version upon reading. The current version number is 2001. The format number is 0 for UFD format, 10000 for CBD format, and 20000 for CFD format. The version and format numbers are added together and stored in the first integer record in the file, thus the first integer will be 2001 for UFD format, 12001 for CBD format, and 22001 for CFD format ZRD files.

The ZRD Uncompressed Full Data Format (UFD)

The first record in the file is a single 32-bit integer indicating the version and format number of the ZRD file. The second record in the file is a single 32-bit integer indicating the maximum possible number of segments in the remaining records. The actual number of segments will always be less than or equal to this number.

The remainder of the file consists of a 32-bit integer indicating the number of segments that follows, then that many segments, then another 32-bit integer indicating the number of segments in the next record, and so on until the end of the file. A single ray segment is described by a "C" structure with the following format:

```
typedef struct
        {
        unsigned int status;
        int level;
        int hit_object;
        int hit_face;
        int unused;
        int in_object;
        int parent;
        int storage;
        int xybin, lmbin;
        double index, starting_phase;
        double x, y, z;
        double l, m, n;
        double nx, ny, nz;
        double path_to, intensity;
        double phase_of, phase_at;
        double exr, exi, eyr, eyi, ezr, ezi;
        } RAYPATH;
```

See "Comments on Data Structure Memory Packing" for more details.

Each data item is described below.

status: bitwise flags indicating the status of the ray. The flags (in integer notation) are: terminated: 1, reflected: 2, refracted: 4, scattered: 8, diffracted: 16, ghost: 32, diffracted from previous object: 64, scattered from previous object: 128, ray segment had fatal error: 256, bulk scattered: 512.

level: The number of ray segments between the ray segment and the original source. This is the number of ray-object intercepts the ray has accumulated.

hit_object: The object number the ray intercepted. If zero, the ray did not hit anything. For the zero segment, this value will be the source segment object number.

hit_face: The face number the ray intercepted. Valid only if hit_object is not zero.

in_object: The object number the ray is propagating inside of. This generally determines the index of refraction the ray is traveling through. A value of zero means inside the "background media".

parent: The (prior) ray segment from which the ray originated. If zero, the ray came from the original source point. Note more than 1 child ray may have the same parent; but each ray has only one parent. The parent segment number will always be less than the child rays segment number.

storage: A temporary buffer used by Zemax for indexing. For segment 0 only, this integer is the ray wavelength number. See nx below for the actual wavelength value. Other segments may contain meaningless data.

xybin, lmbin: The pixel number on a detector object which the ray struck. The xybin is for spatial data, the lmbin is for angular (intensity) data.

index: the index of refraction of the media. Not an adequate description for gradient index media. starting_phase: the initial optical path length the ray starts with. This is usually the contribution from diffraction ratings when rays are split off.

x, y, z: the global coordinates of the ray intercept.

l, m, n: the global direction cosines of the ray. Not an adequate description for gradient index media.

nx, ny, nz: the global normal vector of the object at the intercept point. For segment 0 only, the nx value stores the wavelength of the ray being launched.

path_to: the physical (not optical) path length of the ray segment. intensity: the intensity of the ray segment.

phase_of: the phase of the object. This is the optical phase path length equivalent in lens units added by gratings, holograms, and other phase modifying surfaces.

phase_at: the accumulated total phase of the ray (modulo 2*pi). This value includes accumulated phase due to propagation as well as phase introduced by objects that impart phase (i.e. gratings, etc.). The phase introduced by coatings is not included in this value and is generally considered a "phase aberration". This value is only computed at detector surfaces.

exr, exi, eyr, eyi, ezr, ezi: the electric field in the global x, y, and z coordinates, real and imaginary parts. These values will all be zero if the ray trace did not use polarization.

The ZRD UFD format is subject to change, however, Zemax will automatically convert older formats to the current format. Note the structure size is 208 bytes for each ray segment. If 100,000 rays are traced with an average of 50 segments each, the resulting file will be about 100,000 x 50 x 208 bytes = 1Gb in size. The file sizes can quickly become enormous, so great care must be taken to limit the number of segments before deciding to save the ray data to a file. Use of a filter string to limit the saved data to the rays of interest may substantially reduce the amount of data written to the file.

The ZRD file consists of alternating blocks of data. For example, if three rays are traced, with 5, 7, and 9 segments each, then the ZRD file will be formatted as follows:

```
4 bytes: version
4 bytes: maximum number of segments possible, will be 9 or larger in this example
4 bytes: 5 (the number of segments in ray 1)
208 bytes x 5 (the segment data for ray 1)
4 bytes: 7 (the number of segments in ray 2)
208 bytes x 7 (the segment data for ray 2)
4 bytes: 9 (the number of segments in ray 3)
208 bytes x 9 (the segment data for ray 3)
```

For information on what Zemax can do with ZRD files, see "The Ray Database Viewer".

Comments on Data Structure Memory Packing

Note that typical "C" structures in Windows are packed on 8 byte boundaries. This packing means that 2-byte members are aligned on a memory address that is an integer multiple of 2, 4-byte members are aligned on multiples of 4, and 8-byte members are aligned on multiples of 8. This packing speeds memory access but can create some small gaps in the structures. That is why structure sizes can be slightly larger than the sum of the byte sizes of all the members.

The LightningTrace Control

The LightningTrace Control is used to launch and trace a specific set of rays from a discrete mesh for analysis using defined sources, objects, and detectors. The controls on the dialog box are described below.

For more information, see the "Lightning Trace" tool in the Analyze Tab in-help file.

The Ray Database Viewer

The Ray Database Viewer reads in a previously saved ZRD file, and displays the data in a text format. For information on creating a ZRD file, see the previous section.

For more information, see the "Ray Database Viewer" analysis tool in the Analyze Tab in-help file.

The Detector Viewer

The Detector Viewer displays data from detectors in either a graphic or text format. For information on tracing rays to record data on detectors, see "The Ray Trace Control"

For more information, see the "Detector Viewer" tool in the Analyze Tab in-help file.

Comments regarding Detector Surface objects

Detector Surface objects use a collection of triangles to approximate a curved surface that can detect flux and irradiance. The display of the detected data requires conversion of the triangles into square pixels. This conversion is done automatically when displaying the detector data as a false color, grey scale, or cross section plot. The number of square pixels selected is adjusted to provide a reasonable display resolution, and will vary depending upon the number of triangles. For text listings of Detector Surface objects, the X, Y, Z, R, and Angle data will be listed along with Flux and Irradiance values for each pixel. The X, Y, and Z values are the coordinates of the center of each detector triangle. The R value is the distance from the center of each detector triangle to the object vertex. The Angle values are the angle in degrees to the center of each detector triangle as projected to the XY plane. For more information on defining Detector Surface objects see "Detector Surface object".

The filter string

It is frequently convenient to display only NSC rays or data which have certain properties. For example, when scattering and splitting are turned on, the layout diagram will become very confusing if many rays are traced. The filter string allows definition of a "test" rays must pass before they are drawn or displayed. The filter string syntax consists of logical operations between flags that indicate if one segment within a ray hit, miss, reflected, refracted, scattered, diffracted, or ghost reflected from an object within the NSC group. There are also "Extended" flags which support numerical arguments. The supported flags are described in the table below.

FILTER STRING FLAGS

Flag	Description
_n	Filters that start with an underscore "_" followed by an integer code are defined and used by the Path Analysis feature; see "Path Analysis".
~nnnmmm...[#]	Ray path filter. This filter selects ray branches whose segments follow an explicit path. The first object number defined is the source object, followed by each object the rays interact with, in order. Each object number must be a defined by a three digit integer, with leading zeros added if required. A ray that leaves source object 7, then hits objects 18 and then 9 and is then terminated or hits no other object can be selected using the filter "~007018009". Multiple consecutive object numbers are not redundantly defined; for example, if a ray hits the front face of lens 9 and then hits the back face of the same object 9 need only be listed as a single 009 in the filter definition. Optionally, the filter may be terminated with the # symbol, which indicates any segment that initially follows this path is selected. This allows rays which hit different objects, or no objects at all, after the last object listed are still selected. The maximum number of objects in any one filter is 50.
$nnmm...[#]	Ray path filter alternate form. This is identical to the "~" filter above, except two digit codes are used instead of three digit codes for the object numbers. This is a more convenient form if the number of objects is less than 100.
Bn	Ray bulk scattered inside of object n. If the n value is 0, then bulk scattered segments from any object will return true for this test.
Dn	Ray diffracted after striking object n. See En.
En	Diffracted from parent segment's object n. This flag only gets set for ray segments split from diffractive elements, for order numbers other than zero, when ray splitting is on.
Fn	Scattered from parent segment's object n. This flag only gets set for ray segments split from scattering surfaces when ray splitting is on. The specular segment does not get this flag, only scattered segments. If the n value is 0, then scattered segments from any object will return true for this test.
Gn	Ghost reflected from parent segment's object n. This flag only gets set for ray segments reflected from refractive objects when ray splitting is on. If the n value is 0, then ghost segments from any object will return true for this test.
Hn	Ray hit object n. To test whether a ray hit an object, the flag is of the form Hn. For example, to test if a ray hit object 5, the flag would be H5. See Ln.

Jn	Similar to Gn, except that all segments prior to the ghost reflection point are set to have zero intensity. This allows Detector Viewers to look only at ghost energy, not direct incident energy, even if the ray later ghosted off another object. The zero intensity values will only affect the Detector Viewer, not the ray database viewer or layouts.
Ln	Ray hit object n last. To test whether the last segment of a ray branch hit an object, the flag is of the form Ln. For example, to test if the last segment of a ray branch hit object 5, the flag would be L5. See Hn.
Mn	Ray missed object n. To test whether a ray missed an object, the flag is of the form Mn. For example, to test if a ray missed object 15, the flag would be M15.
On	Ray originated at source number n. O0 (that is "O" as in Origin and "0" as in zero) will select all sources.
Rn	Ray reflected after striking object n. The flag R7 would test if the ray reflected after striking object 7. See Gn.
Sn	Ray scattered after striking object n. This tests the "S" flag as listed in the ZRD file, which refers to scattering at the point a ray strikes an object. See also Fn and X_SCATTER.
Tn	Ray transmitted (refracted) in to or out of object n. The flag T4 would test if the ray refracted in or out of object 4 after striking the object.
Wn	Ray uses wavelength n. If the n value is 0, then rays with any wavelength will return true for this test. Note this filter only tests the <u>initial</u> wavelength for the ray as it leaves the source. If wavelength shifting is used (see the Bulk Scatter section of the Object Properties) the wavelength may change during propagation.
X_AXYG(n,v)	Ray has incident angle (in degrees) on object n in the local x-y plane greater than v. The angle is measured with respect to the +y direction without regard to the direction of propagation. If the ray never strikes object n, this flag is false.
X_AXYL(n,v)	Ray has incident angle (in degrees) on object n in the local x-y plane less than v. The angle is measured with respect to the +y direction without regard to the direction of propagation. If the ray never strikes object n, this flag is false.
X_AXZG(n,v)	Ray has incident angle (in degrees) on object n in the local x-z plane greater than v. The angle is measured with respect to the +z direction without regard to the direction of propagation. If the ray never strikes object n, this flag is false.
X_AXZL(n,v)	Ray has incident angle (in degrees) on object n in the local x-z plane less than v. The angle is measured with respect to the +z direction without

	regard to the direction of propagation. If the ray never strikes object n, this flag is false.
X_AYZG(n,v)	Ray has incident angle (in degrees) on object n in the local y-z plane greater than v. The angle is measured with respect to the +z direction without regard to the direction of propagation. If the ray never strikes object n, this flag is false.
X_AYZL(n,v)	Ray has incident angle (in degrees) on object n in the local y-z plane less than v. The angle is measured with respect to the +z direction without regard to the direction of propagation. If the ray never strikes object n, this flag is false.
X_EXT(n,b)	Ray segment is an extraordinary ray generated from a birefringent interface after the parent ray has hit object n exactly b times. To apply this filter, a search is made for the parent segment that hit object n exactly b times, and only the children of that particular parent segment are considered. If no parent segment hit object n exactly b times, the filter returns false. See also X_ORD.
X_GHOST(n,b)	Ray segment has ghosted exactly b times, and has hit object n at least once. If n is zero, any ray segment that has ghosted b times will pass the test. For example, to consider only all second generation ghosts (ghost rays from ghost parents), use X_GHOST(0, 2).
	X_GHOST does not consider ghost ray segments that end in a TIR condition; although rays that TIR are considered ghosts. For example, if a third generation ghost ray leaves one surface, strikes another surface, and then TIR's from this second surface, X_GHOST(0, 3) will not include this segment because the segment ended in a TIR and not a ray termination (the ray reflected and continued). This same segment will however be included in the filter X_GHOST(0, 4) because the ray ghosted a fourth time (at the TIR point). This is an artifact of how Zemax defines segments and counts ghost rays. In all cases, all ghost rays can be found if sufficiently high values of b are tested. Note rays which TIR from refractive surfaces are considered ghosts, but rays reflected from mirror surfaces are not.
	See also Gn.
X_HIT(n,b)	Ray segment has hit object n exactly b times.
	See also Hn, X_HITS, X_HITFACE, and X_HITFACE2.
X_HITS(n1,n2)	If n2 is zero: Ray has n1 or more hits on any object(s).
	If n2 is not zero, then Ray has between n1 and n2 hits, inclusive.
X_HITFACE(n,f)	Ray segment has hit object n on face f. See also Hn, X_HIT, and X_HITFACE2.
X_HITFACE2(n,f,b)	Ray segment has hit object n on face f exactly b times. See also Hn and X_HIT.

X_IAGT(n,v)	Ray has absolute intensity greater than value v on object n. If the ray never strikes object n, this flag is false.
X_IALT(n,v)	Ray has absolute intensity less than value v on object n. If the ray never strikes object n, this flag is false.
X_IRGT(n,v)	Ray has intensity relative to initial intensity greater than value v on object n. If the ray never strikes object n, this flag is false.
X_IRLT(n,v)	Ray has intensity relative to initial intensity less than value v on object n. If the ray never strikes object n, this flag is false.
X_LGT(n,v)	Ray has local incident x ray direction cosine greater than value v at point on object n. If the ray never strikes object n, this flag is false.
X_LLT(n,v)	Ray has local incident x ray direction cosine less than value v at point on object n. If the ray never strikes object n, this flag is false.
X_MGT(n,v)	Ray has local incident y ray direction cosine greater than value v at point on object n. If the ray never strikes object n, this flag is false.
X_MLT(n,v)	Ray has local incident y ray direction cosine less than value v at point on object n. If the ray never strikes object n, this flag is false.
X_NGT(n,v)	Ray has local incident z ray direction cosine greater than value v at point on object n. If the ray never strikes object n, this flag is false.
X_NLT(n,v)	Ray has local incident z ray direction cosine less than value v at point on object n. If the ray never strikes object n, this flag is false.
X_ORD(n,b)	Ray segment is an ordinary ray generated from a birefringent interface after the parent ray has hit object n exactly b times. To apply this filter, a search is made for the parent segment that hit object n exactly b times, and only the children of that particular parent segment are considered. If no parent segment hit object n exactly b times, the filter returns false. See also X_EXT.
X_SCATTER(n,b)	Ray segment has scattered from parent exactly b times, and has hit object n at least once. If n is zero, any child ray segment split off from the parent ray that has scattered b times will pass the test. For example, to consider only first generation scatter rays, use X_SCATTER(0, 1). This filter tests only the scatter from parent or "F" flag as listed in the ZRD. See also Sn and X_SCATTERF.
X_SCATTERF(n,b)	Ray segment has scattered from object n after the parent of the segment hit object n exactly b times. To apply this filter, a search is made for the parent segment that hit object n exactly b times, and only that particular parent segment is considered. If no parent segment hit object n exactly b times, the filter returns false. For example, to consider only scattered rays that branch off from the parent ray after the third hit on object 5 (that is, the

	ray leaving the source has twice before hit this same object), use X_SCATTERF(5, 3). See also Fn and X_SCATTER.
X_SEGMENTS(n1,n2)	If n2 is zero: Ray has n1 or more segments. If n2 is not zero, then Ray has between n1 and n2 segments, inclusive.
X_WAVERANGE(n, a, b)	Ray has hit object n and has a wavelength between a and b micrometers, inclusive.
X_WAVESHIFT(i,j)	Ray has wave shifted during a bulk scatter event from wavelength i to wavelength j.
X_XGT(n,v)	Ray has local x coordinate greater than value v at point on object n. If the ray never strikes object n, this flag is false.
X_XLT(n,v)	Ray has local x coordinate less than value v at point on object n. If the ray never strikes object n, this flag is false.
X_YGT(n,v)	Ray has local y coordinate greater than value v at point on object n. If the ray never strikes object n, this flag is false.
X_YLT(n,v)	Ray has local y coordinate less than value v at point on object n. If the ray never strikes object n, this flag is false.
X_ZGT(n,v)	Ray has local z coordinate greater than value v at point on object n. If the ray never strikes object n, this flag is false.
X_ZLT(n,v)	Ray has local z coordinate less than value v at point on object n. If the ray never strikes object n, this flag is false.
Z	Ray has fatal error.

All flags except the O, G, E, and F flag indicate what happened at the end of the ray segment, at the point where the segment intercepted the object indicated by the "n" value. The G, E, and F flags indicate that the ray segment's parent struck object "n" and then launched a child ray that was a ghost, diffracted, or scattered segment, respectively. The O flag indicates the ray originated from source object "n". The W flag indicates the ray is of wavelength number "n". The flags of the form X_***(n,v) require two numerical arguments within the parentheses separated by a comma, or a syntax error will be issued.

Each flag is evaluated for each ray traced, and the flag is assigned a status of TRUE or FALSE. The flags may be used alone, or may be combined using logical operands. Logical operands generally act on two other logical flags (the exception is the NOT operand which acts only on the flag to the right of the operand). The supported logical operands are:

&:   Logical AND. Both flags on either side of the & symbol must be TRUE for the AND operation to return TRUE.
|:   Logical OR. If either of the flags are TRUE, OR returns TRUE.
^   Exclusive OR (XOR). If either of the flags are TRUE, but not both, XOR returns TRUE.
!   Logical NOT. Returns TRUE if the flag to the right was FALSE and vice-a -versa.

The parentheses symbols may also be used to set operator precedence.

Filter string examples

To select only rays that hit object 7, the filter string would be:

    H7

To select only rays that hit both objects 7 and 9, the filter string would be:

    H7 & H9

To select rays that must have either a) hit object 7 and object 9, but did not reflect off object 6, or b) missed object 15, the filter string would be:

    (H7 & H9 & !R6) | M15

If only ghost rays generated by striking object 3, 4, or 7 are desired, the filter string would be:

    G3 | G4 | G7

To select only rays incident at an angle of less than 10 degrees on a flat object number 5 whose normal vector points along local +z, the filter string would be (note cos(10) = 0.984808):

    X_NGT(5, 0.984808)

Note only rays which hit the specified object in the X_NGT flag can possibly make the flag true.

The filter string is checked for basic syntax errors, such as mismatched parentheses, but not all possible syntax errors are checked and reported. The number of rays which meet the filter string test may be very small, and perhaps even zero. Zemax still traces the number of rays defined for the source (by the # of layout rays parameter), but only that fraction of these rays which pass the filter will actually be saved, displayed, or drawn. The filter string may also be applied to ray data base files as the ray data is being saved, see The Ray Trace Control. Because filter strings are used with both dynamic and static windows, Zemax does not automatically renumber the filter string arguments when objects are inserted or deleted.

Saving and Loading Detector Data

After a ray trace is performed, detector objects hold data about the energy incident upon the detector. This data may be recreated by "playing back" a ZRD file (see "Ray database (ZRD) files") on the detector using the Detector Viewer (see "The Detector Viewer"). An alternate way of recreating the detector data is to store the data to a file after a ray trace, then loading the detector data back onto the detector object. The Save Detector and Load Detector features are described in "Save Detector Data" and "Load Detector Data".

Detector Data File Formats

Detector data files store energy data for Detector Rectangle, Color, Polar, and Volume objects. The expected file extension for these different detector types is DDR, DDC, DDP, and DDV, respectively. The binary file format for all file types consists of a single header structure, followed by one data block for each pixel in the detector.
Note that when reading detector data files in other programs, there is a gap of 4 bytes that appears between the last int member (i_data) and the first double member (d_data). See "Comments on Data Structure Memory Packing" for more details.

Header Block Format

The header block uses the following format (in C syntax):

```
int version, type, lens_units, source_units, source_multiplier;
int i_data[50];
double d_data[50];
```

version: Currently 1003. Future versions of Zemax may use different version numbers as the format changes, but Zemax will recognize older versions and convert automatically when loading data. Any undefined data values in the format are reserved for future expansion of the format and may be ignored.

type: 1, 2, 3, or 4 for Detector Rectangle, Color, Polar, and Volume objects, respectively.

lens_units: 0, 1, 2, or 3 for mm, cm, in, or M respectively.

source_units: 0, 1, or 2 for Watts, Lumens, or Joules, respectively.

source_multiplier: 0 through 9 for source unit multipliers femto through Tera, respectively.

i_data: integer data that is detector type specific (see below).

d_data: double precision that is detector type specific (see below).

Data for Detector Rectangle And Detector Color Objects:

i_data[0], i_data[1]: number of x pixels, number of y pixels.

i_data[2], i_data[3]: number of rays striking the spatial detector, number of rays striking the angular detector.

d_data[0], d_data[1]: x half width of the detector, the y half width of the detector.

d_data[2] - d_data[5]: angular x minimum, x maximum, y minimum, y maximum.

Data for Detector Polar Objects:

i_data[0], i_data[1]: number of polar pixels, number of angular pixels.

i_data[3]: number of rays striking the detector.

d_data[0], d_data[1]: maximum angle of the detector, radial size of the detector.

Data for Detector Volume Objects:

i_data[0] - i_data[2]: number of x, y, and z direction pixels.

d_data[0] - d_data[2]: half width of the detector in the x, y, and z directions, respectively.

Data Block Format

After the header data is defined and written to the file, the data structures are written. One data block is written for each pixel on the detector. The file formats for the data structures depends upon the detector type as follows.

Data Structure For Detector Rectangle Objects:

double inc_int_pos, inc_int_ang, coh_real, coh_imag, coh_amp;

These are the incoherent flux in position space, the incoherent flux in angle space, and the coherent real, imaginary, and summed amplitude in position space, respectively.

Data Structure For Detector Color Objects:

double pos_P, pos_X, pos_Y, pos_Z, ang_P, ang_X, ang_Y, ang_Z;

The first four are the position space power, tristimulus X, Y, and Z values, and the same 4 values for angle space.

Data Structure For Detector Polar Objects:

double unused1, unused2, unused3, unused4, ang_P, ang_X, ang_Y, ang_Z;

The first four are unused values, followed by the angular space power, tristimulus X, Y, and Z values.

Data Structure For Detector Volume Objects:

double inc_int_pos, inc_absorbed_flux, unused1, unused2, unused3;

These are the incoherent incident flux, the absorbed flux, and three unused values, respectively.

Pixel Ordering

For Detector Rectangle and Color objects, pixel #1 is in the (-x, -y) lower left corner and the subsequent pixels move first across the columns in the +x direction, then up through the rows.

For Detector Polar objects, pixel #1 starts at a polar angle of zero, and subsequent pixels go along the radial arm at zero degrees to the +x axis to the maximum polar angle. The next pixel starts again at polar angle of zero and continues along the radial arm at the first azimuthal angle increment. The pattern repeats for each angular arm along the azimuthal direction.

For Detector Volume objects, pixel #1 is in the (-x, -y) lower left corner of the first plane on the -Z side of the detector, and subsequent pixels move first across the columns in the +x direction, then up through the rows to (+x, +y) of the first plane. The next pixel as at the (-x, -y) corner of the next Z plane, and the pattern continues through all the Z planes.

Special considerations for faceted objects

Ray tracing through faceted objects is generally straightforward in a mathematical sense, with one major exception: how to handle the special case of a ray intercepting "exactly" on the common edge of two adjacent facets. In this context, "exactly" means within the numerical precision limit of the computer. Zemax uses double precision 64 bit numbers for all ray tracing algorithms, and this yields about 12 decimal places of accuracy. However, because of the frequent use of the square root and other math operations, the practical accuracy of many computations is somewhat lower.

The most common difficulty is tracing rays through a roof prism, corner cube, or faceted Fresnel lens where one of the common edges is placed exactly along the x or y axis of the incoming beam. Since the rays hit exactly on the edge of two facets, each with a different normal vector, the reflection or refraction properties become ambiguous.

Physically, infinitely sharp edges cannot exist. Therefore, the solution Zemax uses is to absorb any ray which strikes within the glue distance of an edge of a facet that borders another facet with a different normal vector. An absorbed ray is terminated, and its energy is discarded.

This problem may be avoided by either of these techniques:

For sequential ray tracing, offset the object axis a very small amount in the X or Y directions; so the rays will "see" one facet at a time. An offset of about 1e-6 lens units (which is only 1 nanometer if the lens units are mm) is generally sufficient.

For non-sequential ray tracing, either offset the object as for sequential ray tracing, or use sources with random ray distributions. Occasionally a ray will strike within the glue distance of the facet edge, and the ray energy will be absorbed; but the total amount of energy should be very small.

Generally, using randomly generated ray sets, such as those in the image analysis or dithered spot diagram, will eliminate this problem.

Note that any real roof prism or corner will physically not "correctly" reflect or refract rays within very small distances of the edge due to manufacturing limitations and physical optics interference.

Comments about DLLs

Zemax supports several different types of DLLs for extending the capabilities of the non-sequential features, including DLLs for sources, gradient index media, diffraction, and scattering.

To create a user defined DLL source, a suitable compiler or development tool that can generate 32 bit Windows compatible DLLs is required. It is also assumed that the user can write the required code, and most importantly, ensure that the code is reliable and bug-free. To maximize speed, Zemax performs very little error checking on data returned by the DLL, and so buggy DLLs are quite capable of bringing Zemax to a crash.

For this reason, technical support on the implementation of DLLs is strictly limited to illustrating that the provided sample files work correctly. If you need a DLL, and do not possess the desire or ability to write them yourself, please feel free to contact Radiant Zemax LLC for a quote on developing a custom DLL to meet your requirements. Radiant Zemax LLC has considerable experience in developing algorithms, and can generally write DLLs at competitive rates on short notice.

The best way to learn the use of the various DLL types is to study an existing DLL. The sample DLLs provided with Zemax include extensive documentation and comments on the data format; see any of the sample source code files for examples.

Non-Sequential Geometry Objects

Zemax NSC object types include ellipses, triangles, rectangles, spheres, cylinders, and other basic shapes. Complex objects such as arbitrary prisms, aspheric lenses, torics, toruses, and other optical components are also available. The reflective, refractive, and absorptive properties of these objects are determined by the material assigned to the objects. For details on reflective, refractive, and absorptive properties, see the Technical Reference sections called "Using Glass Catalogs" and "Polarization Analysis".

Each NSC object type is described in the following summary table, and in greater detail in the following sections. Note these basic objects may be combined to form more complex objects. See the "Object Placement" section of the Technical Reference article "Non-sequential Overview" for information on placing objects inside or adjacent to one another.

For information on sources and detectors, please see the Technical Reference articles called "Non-Sequential Sources" and "Non-Sequential Detectors".

If an object type is required that is not listed, please email technical support to suggest the new object type be added to Zemax.

Summary of NSC Objects

Annular Aspheric Lens	An annular volume with an aspheric surface on one face.
Annular Axial Lens	An annular lens formed by sweeping an aspheric cross section about a decentered Z axis.
Annular Volume	A volume formed by two tilted, elliptical annular planar faces.
Annulus	A planar ellipse with an elliptical hole in the middle.
Array	This object creates a 3D parallelogram array of objects based upon a master object.
Array Ring	This object creates a ring array of objects based upon a master object.
Aspheric Surface	A surface with a conic substrate plus even and odd radial polynomial power aspheric terms.
Aspheric Surface 2	Identical to the Aspheric Surface, with support for more complex aperture types.
Axicon Surface	An axicon formed by rotating a spherical arc around the z axis.
Biconic Lens	A circular or rectangular lens with separate curvature and conic in the X and Y directions on both front and back faces.

Biconic Surface	A surface with separate curvature and conic in the X and Y directions. May be partially hyperhemispheric under special circumstances.
Biconic Zernike Lens	A lens with a standard front face and a biconic Zernike rear face.
Biconic Zernike Surface	A surface with a biconic Zernike shape. The surface supports elliptical, rectangular, and user apertures, and the biconic and Zernike terms may be separately decentered.
Binary 1	A standard lens with a binary 1 phase profile on the front face. The binary 1 phase profile is an X-Y polynomial.
Binary 2	A standard lens with a binary 2 phase profile on the front face. The binary 2 phase profile is a radial polynomial in even powers of r.
Binary 2A	An even aspheric lens (front and back faces) with a binary 2 phase profile on the front face. The binary 2 phase profile is a radial polynomial in even powers of r.
Boolean	A very general object defined by a Boolean combination of other objects.
CAD Assembly: Autodesk Inventor	A dynamic link to an Autodesk Inventor assembly.
CAD Assembly: **Creo Parametric**	A dynamic link to a Creo Parametric assembly.
CAD Assembly: SolidWorks	A dynamic link to a SolidWorks assembly.
CAD Part: Autodesk Inventor	A dynamic link to an Autodesk Inventor part.
CAD Part: Creo Parametric	A dynamic link to a Creo Parametric part.
CAD Part: SolidWorks	A dynamic link to a SolidWorks part.
CAD Part: STEP/IGES/SAT	An object or objects imported from a CAD program in STEP, IGES, or SAT format.
CAD Part: STL	A general polygon object defined in STL format, typically exported from a CAD program.
CAD Part: Zemax Part Designer	An object created using the Zemax Part Designer.
CPC	A compound parabolic concentrator (CPC). The CPC may be hollow or solid.
CPC - Rectangular	A CPC with rectangular symmetry and separate X and Y aperture and acceptance angles.
Cone	A section of a cone defined by two points, r and z, which forms a line segment, which is then rotated around the local z axis.
Cylinder Pipe	A tapered cylindrical surface shape.
Cylinder Volume	A tapered cylindrical volume, with end caps.
Cylinder 2 Pipe	A cylindrical surface shape with tilted ends.
Cylinder 2 Volume	A cylindrical volume, with tilted end caps.
Diffraction Grating	A standard lens with a diffraction grating with constant or variable line spacing on the front face.
Dual BEF Surface	An idealized dual brightness enhancement film surface
Ellipse	A planar ellipse.
Elliptical Volume	An elliptical volume (or shell). The shape is a tapered cylinder with an elliptical cross section.
Even Asphere Lens	A rotationally symmetric lens with up to 16th order aspheres on both the front and back faces.
Extended Odd Asphere Lens	A lens with extended odd asphere surfaces on the front and back faces.
Extended Polynomial Lens	A lens with extended polynomial surfaces on front and back faces.
Extended Polynomial Surface	A surface with an extended polynomial shape.

Extruded	A solid volume formed by extruding a User Defined Aperture.
Faceted Surface	A standard surface faceted into flat polygons.
Freeform Z	A solid lens or shell defined by rotating a spline curve around the Z axis. The spline curve is defined by points along the Z axis.
Fresnel 1	A true Fresnel lens, either radial or cylindrical, defined in terms of a number of construction parameters.
Fresnel 2	An idealized Fresnel lens object having a circular or rectangular shape, with one face an ideal Fresnel lens with either radial or cylindrical polynomial aspheric terms over a base conic asphere.
Grid Sag Lens	A lens with a standard front face and grid sag rear face.
Grid Sag Surface	A surface defined by a table of sag points.
Hexagonal Lenslet Array	A lenslet array with hexagonal symmetry.
Hologram Lens	A circular or rectangular solid with an optically fabricated hologram on the front face.
Hologram Surface	A circular or user defined plane, conic, spherical or aspherical surface with an optically fabricated hologram on the face.
Jones Matrix	An elliptical flat object which supports 8 ABCD parameters which can be use to model neutral density filters, polarizers, rotators, and other arbitrary Jones Matrix type objects.
Lenslet Array 1	A rectangular shaped lens array object with a curved back face. This object may be diffractive.
Lenslet Array 2	A rectangular shaped lens array object with curved front and back faces.
Micro Electro Mechanical System (MEMS)	An array of flat faces (intended to be made of material "MIRROR") where the individual faces may be tipped about any of 3 angles. The faces may be addressed either by rows, by columns, or by individual pixels.
Null Object	A non-existent object. This is the default when new objects are inserted into the NSC Editor, and is handy to use as a place holder.
Odd Asphere Lens	A rotationally symmetric lens with up to 12th order odd and even power aspheres on both the front and back faces.
Paraxial Lens	An idealized planar surface that acts like a thin lens with separable powers.
Polygon Object	A user defined general polygon object which may be open or closed, and may have refractive, reflective, or absorptive faces, or a mixture of these types of faces.
Ray Rotator	An imaginary component that rotates incident rays in two directions.
Rectangular Corner	A corner composed of 3 rectangular surfaces.
Rectangle	A rectangle defined by a width and height.
Rectangular Pipe	A four sided box.
Rectangular Pipe Grating	A four sided box with a grating on the side faces.
Rectangular Roof	A roof composed of two rectangular surfaces meeting at an angle.
Rectangular Torus Surface	A section of a rectangular torus which defines a surface; open at the ends.
Rectangular Torus Volume	A section of a rectangular torus with closed ends to define a volume.
Rectangular Volume	A six sided box enclosing a volume.
Rectangular Volume Grating	A six sided box with a diffraction grating on side faces.
ReverseRadianceTM Detector	A detector used with the ReverseRadianceTM feature.

ReverseRadianceTM Target	A target object used with the ReverseRadianceTM feature.
Slide	A color RGB file, in either BMP, JPG, or PNG format. The slide may be scaled to any size and aspect ratio, and may be used to filter a source to create colored image sources.
Sphere	A sphere.
Standard Lens	A 5 surface lens object with two Standard faces, two flat edges, and a cylindrical outer edge.
Standard Surface	A surface with the same shape as the Zemax Standard surface (a plane, sphere, conic asphere, or hyperhemisphere).
Swept Object	A surface or volume formed by sweeping the cross section of another object around an arbitrary axis through an angle.
Tabulated Faceted Radial	A faceted object generated from a table of coordinates rotated about the local z axis. This object type can be used to define surfaces or volumes.
Tabulated Faceted Toroid	A faceted object generated from a table of coordinates rotated about an axis parallel to the local y axis. This object type can be used to define surfaces which are cylindrical or toroidal.
Tabulated Fresnel Radial	A smooth Fresnel lens generated from a table of coordinates. This is very similar to the Tabulated Faceted Radial object except the surfaces are smooth, not faceted. See also the Fresnel object.
Toroidal Hologram	A circular, elliptical, or rectangular lens with aspheric toroidal surfaces on the front and back sides, and a hologram on the front face.
Toroidal Lens	A circular, elliptical, or rectangular lens with aspheric toroidal surfaces on the front and back sides.
Toroidal Surface	A rectangular surface with an aspheric toroidal shape.
Torus Surface	A section of a torus which defines a surface; open at the ends.
Torus Volume	A section of a volume torus; like a surface but with end caps to enclose a volume.
Triangular Corner	A corner composed of 3 triangles.
Triangle	A triangle defined by 3 points in a plane.
User Defined Object	An object defined by a user provided DLL.
Wolter Surface	A radially symmetric shell object used in x-ray telescopes.
Zernike Surface	A surface defined by radial aspheric and standard Zernike polynomial sag terms.

Annular Aspheric Lens
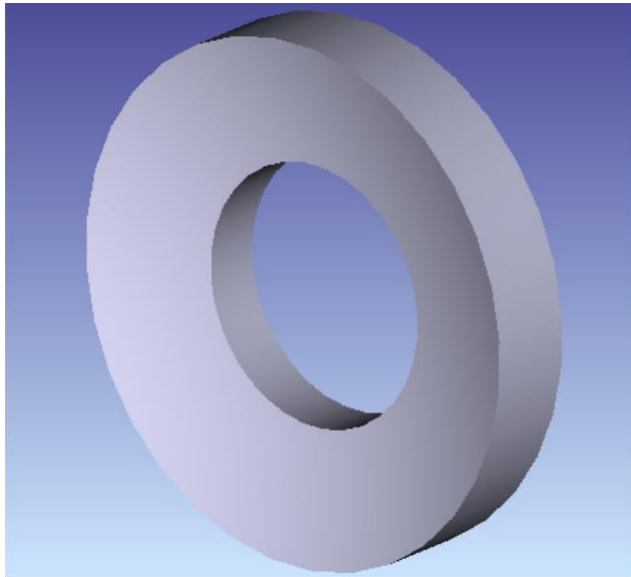
The annular aspheric lens is an annular solid with a conic asphere with even aspheric polynomial terms to 16th order, identical to the Even Asphere Lens object, on both front and back faces. See "Even Asphere Lens" for a description of the even aspheric polynomial.

The annular shape is defined by a min and max radial aperture which may be different on the front and back faces. The aspheric rear face is offset from the front by a thickness measured at a specified radial aperture, which may be zero.

For backward compatibility with the original definition of this object, the back face parameters are listed before the front surface parameters.

The following restrictions apply: all apertures must be positive, maximum apertures on each face must be larger than the minimum aperture on that face, and if one face has zero for a minimum aperture the other face must also have a zero minimum aperture.

The object is defined by the following parameters:

1-2: The Minimum and Maximum Radial Apertures on the front face.

3-4: The Minimum and Maximum Radial Apertures on the back face.

5: The Thickness Aperture, which is the radial aperture used to measure the object thickness. This value may be zero.

6: Back face radius of curvature.

7: Back face conic.

8: Thickness (measured at the Thickness Aperture).

9-16: The Even Aspheric coefficients α_1 through α_8 for the back face.

17: Front face radius of curvature.

18: Front face conic.

19-26: The Even Aspheric coefficients α_1 through α_8 for the front face.

Face Numbers: Outside face connecting the "maximum" apertures Face 0, front face Face 1, back face Face 2, inside face connecting the "minimum" apertures on the front and back Face 3.

Annular Axial Lens
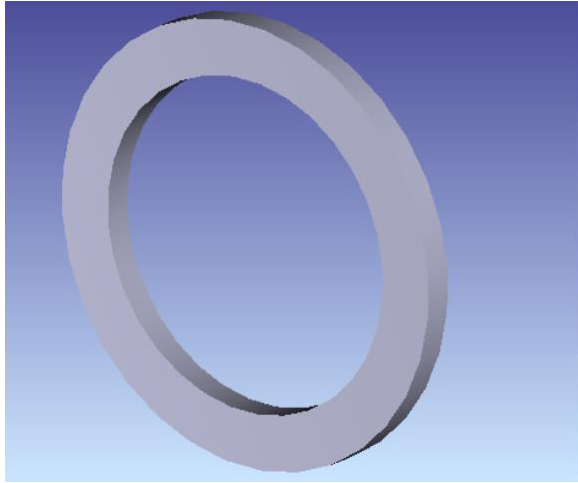
The annular axial lens is defined by sweeping the cross section of a bi-aspheric lens around a decentered axis parallel to the axis of symmetry of the cross section. The cross section consists of two aspheric faces, each of which is a polynomial asphere. The cross section has the same shape as the Even Asphere Lens, see "Even Asphere Lens" . The cross section may however be decentered along the radial direction. The defining parameters are:

1: The radius of curvature of the front face, use zero for flat.

2: The conic constant of the front face.

3: The radius of curvature of the rear face, use zero for flat.

4: The conic constant of the rear face.

5: The thickness of the cross section. The thickness must be large enough so that the edge thickness of the part is at least 0.001 times the center thickness.

6: The radial aperture of the cross section.

7: The radius of rotation from the local Z axis to the axis of symmetry of the cross section. This value must be

at least 1% larger than the radial aperture of the cross section defined by parameter 6.

8: The decenter of the cross section within the lens aperture. Positive decenters shift the axis of the cross

section radially outward, negative decenters shift the axis radially inward.

9: Unused.

10-17: The front face coefficients . α_1 through α_8

18-19: Unused.

20-27: The rear face coefficients . α_1 through α_8

Face Numbers: Outside face connecting the "maximum" apertures Face 0, front face Face 1, back face Face 2, inside face connecting the "minimum" apertures on the front and back Face 3.

Annular Volume
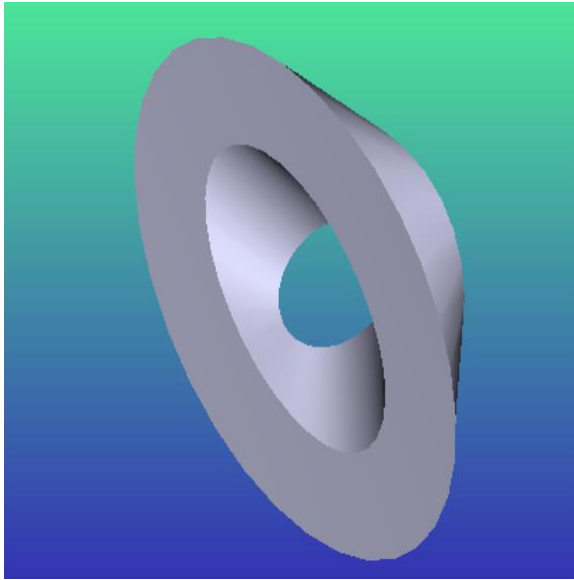
The annular volume consists of tilted annular elliptical faces front and rear, separated by an axial distance. The front and rear face minimum and maximum coordinates are measured on the XY plane. These coordinates form an ellipse on the front and rear reference planes. The inner and outer elliptical cones formed by connecting the front and rear elliptical shapes are the inner and outer surfaces of the annular volume. The front and rear tilted faces are then tilted by the specified angles in x and y. The boundary shape of each face is defined by the intersection of the tilted plane and the annular elliptical cone. The defining parameters are:

1-2: Unused

3: The front face minimum x coordinate.

4: The front face minimum y coordinate.

5: The rear face minimum x coordinate.

6: The rear face minimum y coordinate.

7: The front face maximum x coordinate.

8: The front face maximum y coordinate.

9: The rear face maximum x coordinate.

10: The rear face maximum y coordinate.

11: The length along the local Z axis.

12: The front face x tilt angle in degrees.

13: The front face y tilt angle in degrees.

14: The rear face x tilt angle in degrees.

15: The rear face y tilt angle in degrees.

Face Numbers: Outside face connecting the "maximum" apertures Face 0, front face Face 1, back face Face 2, inside face connecting the "minimum" apertures on the front and back Face 3.

Annulus
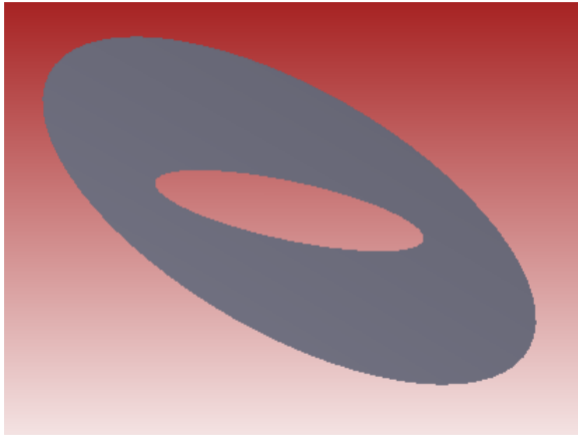
The annulus is a plane elliptical surface shape defined by 4 parameters:

1: The X Maximum Half Width.

2: The Y Maximum Half Width.

3: The X Minimum Half Width.

4: The Y Minimum Half Width.

The annulus resides entirely within the local XY plane. If the minimum half widths are set to zero, then no "hole" exists and rays will hit anywhere within the maximum half widths. In this case, the annulus is identical to the ellipse.

The reference coordinate is the center of the annulus. Face Numbers: All faces Face 0.

Array
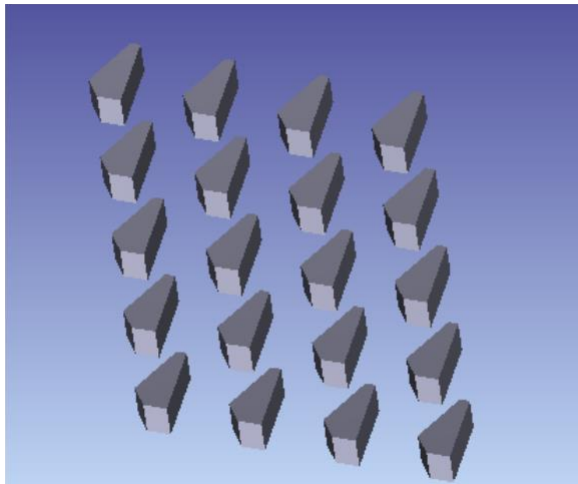
This object is a collection of identical objects, based upon a single parent object, arranged in a 3D array. The individual array elements are all functionally identical to the parent object, just as if a copy of the parent object had been placed in each array location. The 3 Axes of the array need not be orthogonal, and are user definable; see the discussion below. The Array object is generally faster to trace and more memory efficient than the equivalent system created using the

replicate tool (see "Replicate Object") or using an array of similar objects as supported by the Lenslet 1 and Lenslet 2 objects. The Array object is defined by the following parameters:

1: The parent object #. The parent object must precede the array object in the NSC Editor.

2-4: The number of array elements in the X', Y', and Z' directions

5-7: The element spacing in the X', Y', and Z' directions in lens units. See the discussion for non-uniform spacing.

8-10: The direction cosines of the X' axis of the array.

11-13: The direction cosines of the Y' axis of the array.

14-16: The direction cosines of the Z' axis of the array.

17-19: The element tilt in degrees about the X, Y, and Z directions. Note these tilts are about the "normal" local X, Y, and Z axes, and not the user defined X', Y', and Z' axes.

20: Draw Limit. If the product of the number of elements in each direction (Nx x Ny x Nz) exceeds the Draw Limit, the individual array elements will not be drawn. Instead, a simple box that approximates the size of the array is drawn. This feature is intended to speed up the rendering of the array object when the number of elements is large.

21: Draw Boundary: If this is set to 1, the box described in parameter 20 above will be drawn in all cases. This feature is useful to compare the box dimensions and orientation with the actual array elements; which may extend beyond the box boundaries.

22-30: The non-linear spacing coefficients, see the discussion below.

31-33: The maximum dimensions of the array in the X', Y', and Z' directions. These values are not set by the user. The values are computed based upon the spacing coefficients and the number of elements defined. The optimization operands NPGT, NPLT, and NPVA may be used to constrain the overall size of the array while optimizing the non-linear spacing coefficients. The maximum dimensions reported in these parameter values is the center-to-center spacing from the first to the last array element in each direction, and does not consider the size of the individual array elements.

The parent object may be any object type, except for a source, detector, or another array object. If the parent object has the "Object Is A Detector" box checked, this property will be ignored in the array elements. To create an array of sources, see "Modeling arrays of sources" .

The number of array elements in the 3 directions may in principle be any number, however, very large arrays may require excessive computation time or computer memory to ray trace or render (see parameters 20 and 21 for settings to eliminate the rendering of large arrays).

The array elements are arranged on a 3D grid where the grid axes are not necessarily the usual orthogonal XYZ axes. The X', Y'. and Z' axes default to the normal orthogonal local axes, however, any non-parallel orientation for the axes may be defined by specification of the direction cosines for the primed axes. The array positions are computed for the element with index ijk (i for X', j for Y', k for Z') according to the following expression:

$$\vec{r} = \begin{bmatrix} i\delta_x \\ j\delta_y \\ k\delta_z \end{bmatrix}, \text{ where}$$

$$\begin{bmatrix} i\delta_x \\ j\delta_y \\ k\delta_z \end{bmatrix} = \begin{bmatrix} X_x & Y_x & Z_x \\ X_y & Y_y & Z_y \\ X_z & Y_z & Z_z \end{bmatrix} \begin{bmatrix} i\delta_x \\ j\delta_y \\ k\delta_z \end{bmatrix}$$

and the Xx, Xy, Xz values are cosines defining the X' axis (parameters 8, 9, and 10), and similarly for Y' and Z', and the Δ_1 values are the element spacings along the axes (parameters 5, 6, and 7). Note that the direction cosine vectors are always normalized automatically. Once the array element is located, the local rotation about the X, Y, and Z axis is performed, in a manner identical to the usual coordinate rotations (see "Object Placement").

The Δ_1 values in the three directions are used to define a uniform spacing of the array elements. However, a non-uniform spacing may also be defined using parameters 22-30. The position of the array elements along the X' direction is in general defined as

$$x_i = \Delta_1 i + \Delta_2 i^2 + \Delta_3 i^3 + \Delta_4 i^4$$

where Δ_1 is the uniform spacing (defined by parameters 5, 6, and 7 for X', Y', and Z', respectively), the index "i" goes from 0 to n_x-1 (n_x is the number of elements in the X' direction), and the Δ_k coefficients are the kth order terms measured in lens units. If any of the Δ_k coefficients are non-zero, the array spacing will be non-uniform. A similar expression and set of coefficients is available for the Y' and Z' directions. The net spacing between any two adjacent elements cannot be negative. Be careful not to define the non-linear coefficients in such a way that the spacing between the elements changes sign causing the array to fold back on itself.

Most properties of the Array object are defined by the parent object. The exceptions, which may be distinct for the array object and the parent object, are the "Rays Ignore This Object" and the "Do Not Draw Object" settings. All other properties, including face definition, material, coatings, scattering, diffraction, or gradient index proper- ties, are as defined for the parent object, and the array object settings will be ignored. If the parent object is not needed, use the "Rays Ignore This Object" and "Do Not Draw Object" settings on the parent object (See "Type tab" and "Draw tab"). The array will still be created and ray traced even if the parent object is not drawn or ray traced. Note that rays striking an array object will report hitting the parent object number, not the array object number. For this reason, when using ZRD files, optimization operands, or filter strings, the object number used to refer to ray data should be the parent object number.

When nesting array objects inside or outside other objects, the usual nesting rules apply. For more information see "Nesting volumes" . However, it is the placement of the parent object in the NSC Editor, and not the array object, that determines the nesting rules. Specifically, if an array object is to be nested inside of another, outermost object, the parent object must follow the outermost object in the editor.

Array Ring
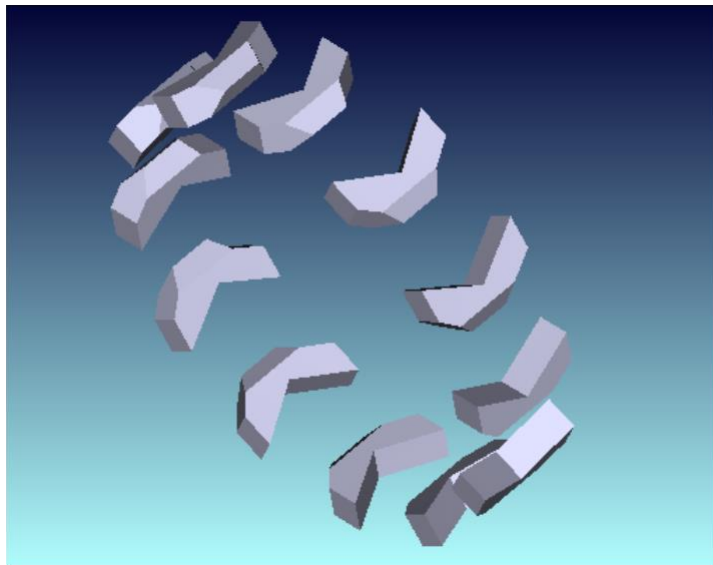
This object is a collection of identical objects, based upon a single parent object, arranged in a ring or arc. The individual array elements are all functionally identical to the parent object, just as if a copy of the parent object had been placed in

each array location. The Array Ring object is generally faster to trace and more memory efficient than the equivalent system created by copying the parent object to each element location. The Array Ring object is defined by the following parameters:

1: The parent object #. The parent object must precede the array object in the NSC Editor.

2: The mode number.

3: The number of array elements, N.

4: The Radius of the ring or the reference sphere, R, in lens units.

5-9: The parameters α , β , γ , δ , and ε . Whenever these parameters are angles, the units are in degrees.

The parent object may be any object type, except for a source or detector. If the parent object has the "Object Is A Detector" box checked, this property will be ignored in the array elements. To create an array of sources, see "Modeling arrays of sources" .

The number of array elements may in principle be any number, however, very large arrays may require excessive computation time or computer memory to ray trace or render.

The array elements are arranged on a ring (or arc). There are two ways to define the positions of the array elements, defined by the mode flag. Mode 0 results in a ring of objects in a plane. Mode 1 creates a ring of objects on a reference sphere. The method for placing each array elements is as follows for each mode:

Mode 0: The parent object is first placed at the vertex of the Array Ring. The object is then rotated around the array local +Z axis by the angle α. The object is then shifted along the array local +X axis by the distance R. The object is then rotated around the array local Z axis by the angle γ. This is the position of the first element in the array. Subsequent array elements are further rotated around the array local Z axis by an angle δ /N. The angular position of the array element "k" is given by θ = γ + k* δ /N, where k varies from 0 to N-1. For a complete ring, δ should be 360. Mode 0 may also be used to form a Spiral of Archimedes, where R is replaced by R + k* β * δ /N and β is interpreted to be in lens units per degree. The value ε in lens units may also be used to define a step change in R every time the angle given by k* δ /N passes 360 degrees. Ray tracing will generally be much slower when either β or ε are not zero, since the symmetry of the array is broken.

Mode 1: The parent object is first placed at the vertex of the Array Ring. The object is then rotated around the array local +Z axis by the angle α . The object is then rotated an angle β about an axis parallel to the array local Y axis that is shifted by R in the array local +Z direction, so the array element's local Z points toward the center of a sphere with radius R (or away from the center if R is negative). The sign convention for the angle β used here is that β is always positive, and the object is always rotated toward the local +X axis, whether or not R is positive or negative. The object is then rotated around the array local Z axis by the angle γ. This is the position of the first element in the array (k = 0). Subsequent array elements are further rotated around the Z axis by an angle δ /N.

For all modes, to form a complete ring, δ should be 360. Smaller values of δ will form an arc of objects. Larger values of δ may cause elements to overlap, which may create an invalid geometry. The angle γ "clocks" the starting point of the ring or arc around Z.

Most properties of the Array Ring object are defined by the parent object. The exceptions, which may be distinct for the Array ring object and the parent object, are the "Rays Ignore This Object" and the "Do Not Draw Object" settings. All other properties, including face definition, material, coatings, scattering, diffraction, or gradient index properties, are as defined for the parent object, and the array object settings will be ignored. If the parent object is not needed, use the "Rays Ignore This Object" and "Do Not Draw Object" settings on the parent object (See "Type tab" and "Draw tab"). The array will still be created and ray traced even if the parent object is not drawn or ray traced. Note that rays striking an array object will report hitting the parent object number, not the array object number. For this reason, when using ZRD files, optimization operands, or filter strings, the object number used to refer to ray data should be the parent object number.

When nesting array objects inside or outside other objects, the usual nesting rules apply. For more information see "Nesting volumes" . However, it is the placement of the parent object in the NSC Editor, and not the array object, that determines the nesting rules. Specifically, if an array object is to be nested inside of another, outermost object, the parent object must follow the outermost object in the editor.

Aspheric Surface
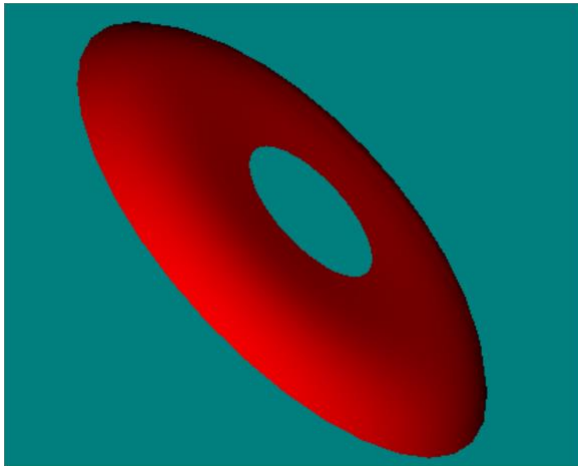
An aspheric surface is defined by the following sag equation:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=1}^{N} \alpha_i r^i$$

where c is the curvature of the surface, k is the conic constant, r is the radial coordinate, and the α terms are aspheric coefficients. The surface supports specification of both a minimum and a maximum radial aperture; so annular surfaces may be defined. Note that both even and odd terms are defined; up to approximately 240 coefficients may be used.

The following parameters are used to define the aspheric surface:

1: The radius of curvature. If this value is zero, then the curvature is assumed to be zero.

2: The conic constant k.

3: The maximum radial aperture in lens units.

4: The minimum radial aperture in lens units. This value may be zero.

5-6: Unused.

7: The number of terms to use in the aspheric expansion. Ray tracing will be faster if this term is no larger than the highest order non-zero coefficient.

8-249: The α coefficients on the polynomial expansion.

Face Numbers: All faces Face 0. This object supports user defined apertures, see "User defined apertures" . For other aperture shape options, see "Aspheric Surface 2" .

Aspheric Surface 2
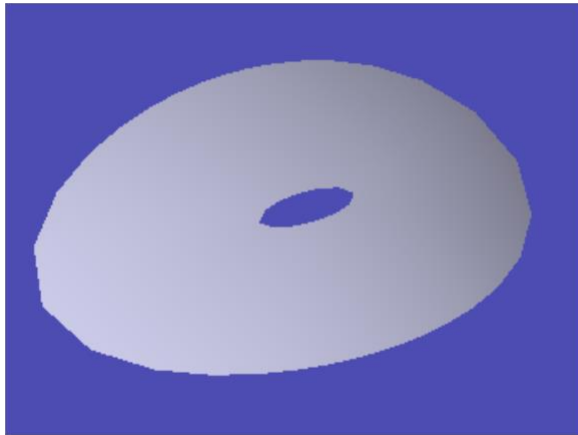
This surface is similar to the Aspheric Surface above, with added support for elliptical or rectangular decentered annular apertures. The following parameters are used to define the aspheric surface:

1: The radius of curvature. If this value is zero, then the curvature is assumed to be zero.

2: The conic constant k.

3: The maximum X direction aperture in lens units.

4: The maximum Y direction aperture in lens units.

5: The minimum X direction aperture in lens units. This value may be zero.

6: The minimum Y direction aperture in lens units. This value may be zero.

7: The X direction aperture decenter in lens units.

8: The Y direction aperture decenter in lens units.

9: The "Is Rectangle" flag. Use 0 for elliptical symmetry, 1 for rectangular symmetry.

10-11: Unused.

12: The number of terms to use in the aspheric expansion. Ray tracing will be faster if this term is no larger than the highest order non-zero coefficient.

13-249: The α coefficients on the polynomial expansion.

Face Numbers: All faces Face 0.

Axicon Surface
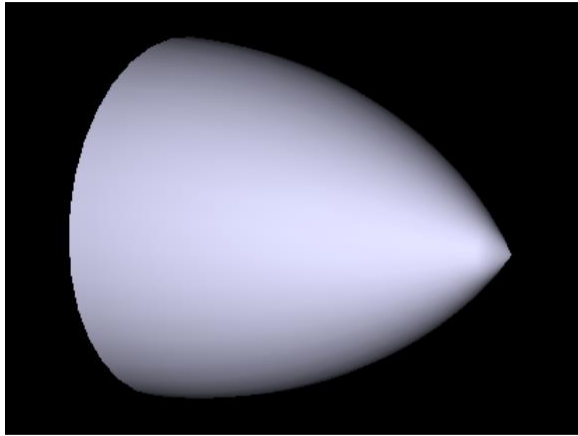
There are many types of axicons. Common polynomial types can be modeled using the Aspheric Surface (see "Aspheric Surface").

The Axicon Surface object uses a different method for defining the object shape. The Axicon surface is defined by a section of a circle (an arc) that lies in the YZ plane. The arc is defined by three points. The start of the arc is at (z=0, y=0). The center of the arc is at the user defined coordinates (cz, cy). The end of the arc is at the point where z = L, where L is a user defined parameter corresponding to the length of the axicon along the Z axis. The surface is formed by revolving the arc around the Z axis. If cz is positive, the negative of the absolute value of cy is taken, and the portion of the circle lying in the first quadrant is used to define the surface. If cz is negative, the absolute value of cy is taken, and the portion of the circle lying in the first quadrant is used to define the surface.

The following parameters are used to define the axicon surface:

1: The length of the axicon along the local z axis.

2: The z coordinate of the center of the arc measured in the YZ plane.

3: The y coordinate of the center of the arc measured in the YZ plane.

Face Numbers: All faces Face 0.

Biconic Lens
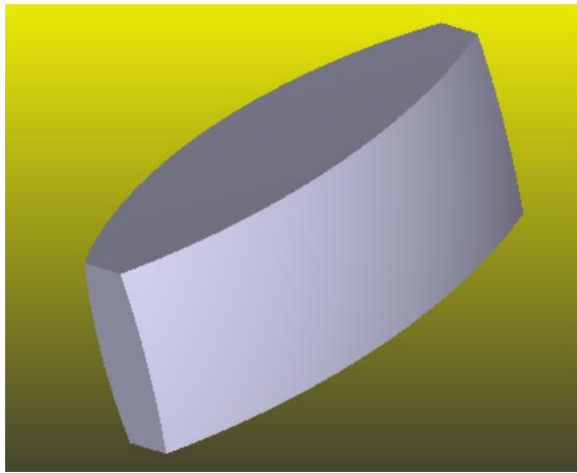
A biconic lens is similar to a toroidal lens, except the conic constant and base radius may be different in the X and Y directions. The biconic lens allows specification of Rx, Ry, Kx, and Ky for the front and back surfaces independently. The sag of a biconic surface is given by:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x^2 x^2 - (1 + k_y)c_y^2 y^2}}$$

where

$$c_x = \frac{1}{R_x}, \; c_y = \frac{1}{R_y}$$

The biconic lens is defined by these parameters:

1: The radial height of the lens object in lens units. This value is used for the y direction half height if the lens is rectangular.

2: The x half width of the lens object in lens units. If zero, the lens is circular; otherwise, the lens is rectangular.

3: The center thickness of the lens in lens units.

4-5: Unused.

6: The base radius of curvature in the XZ plane for the front surface. If this value is zero, then the XZ curvature is assumed to be zero.

7: The base radius of curvature in the YZ plane for the front surface. If this value is zero, then the YZ curvature is assumed to be zero.

8: The X direction conic for the front surface.

9: The Y direction conic for the front surface.

10: The base radius of curvature in the XZ plane for the back surface. If this value is zero, then the XZ curvature is assumed to be zero.

11: The base radius of curvature in the YZ plane for the back surface. If this value is zero, then the YZ curvature is assumed to be zero.

12: The X direction conic for the back surface.

13: The Y direction conic for the back surface.

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Biconic Surface

A biconic surface is similar to a toroidal surface, except the conic constant and base radius may be different in the X and Y directions. The biconic surface allows specification of Rx, Ry, Kx, and Ky directly. The sag of a biconic is given by:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x{}^2 x^2 - (1 + k_y)c_y{}^2 y^2}}$$

where

$$c_x = \frac{1}{R_x}, \; c_y = \frac{1}{R_y}$$

The biconic surface is defined by 13 parameters:

1: The base radius of curvature in the XZ plane. If this value is zero, then the XZ curvature is assumed to be zero.

2: The base radius of curvature in the YZ plane. If this value is zero, then the YZ curvature is assumed to be zero.

3: The X direction conic.

4: The Y direction conic.

5-6: The maximum X/Y aperture in lens units.

7-8: The minimum X/Y aperture in lens units. These parameters are ignored if the "Is Rectangle?" parameter is zero.

9-10: Unused.

11: The "Is Rectangle?" flag. If zero, the resulting surface shape will have an elliptical boundary. If non-zero, the surface will have a rectangular boundary.

12: The "Is Top Hyper?" flag. If zero, the maximum Y aperture will lie on the non-hyperhemispheric portion of the surface. If non-zero, the maximum Y aperture of the surface will be hyperhemispheric. This parameter is ignored if the

"Is Rectangle?" parameter is zero, if the Y direction conic is less than or equal to -1.0, if the maximum Y aperture is less than zero, or if the XZ base radius is non-zero.

13: The "Is Bot Hyper?" flag. If zero, the minimum Y aperture will lie on the non-hyperhemispheric portion of the surface. If non-zero, the minimum Y aperture of the surface will be hyperhemispheric. This parameter is ignored if the surface is elliptical, if the Y direction conic is less than or equal to -1.0, if the minimum Y aperture is greater than zero, or if the XZ base radius is non-zero.

Making a hyperhemispheric surface
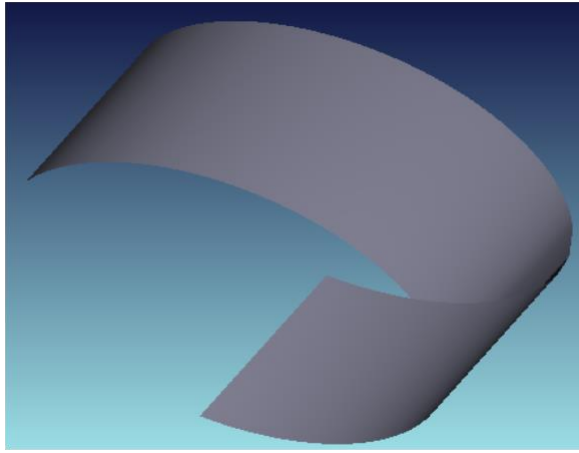
The biconic surface can also be used to make a hyperhemispheric cylinder surface, but only under these conditions:

1) The Rx and Kx values are set to zero.

2) The "Is Rectangle?" flag is set to 1 (that is, the surface boundary is a rectangle).

3) The Ky value is greater than -1.

For a surface that is hyperhemispheric on the top half, set the maximum Y aperture to a positive value and set the "Is Top Hyper?" flag to unity.

For a surface that is hyperhemispheric on the bottom half, set the minimum Y aperture to a negative value and set the "Is Bot Hyper?" flag to unity.

The surface may be hyperhemispheric in both the top and bottom portions at once, but currently only for the case of Rx = 0.0. If either the top or bottom is hyperhemispheric, the actual maximum Y aperture is given by

$$y^2 = \frac{R_y{}^2}{1 + k_y}$$

In this special case, a hyperhemispherical cylinder will result.

Face Numbers: All faces Face 0.

Biconic Zernike Lens
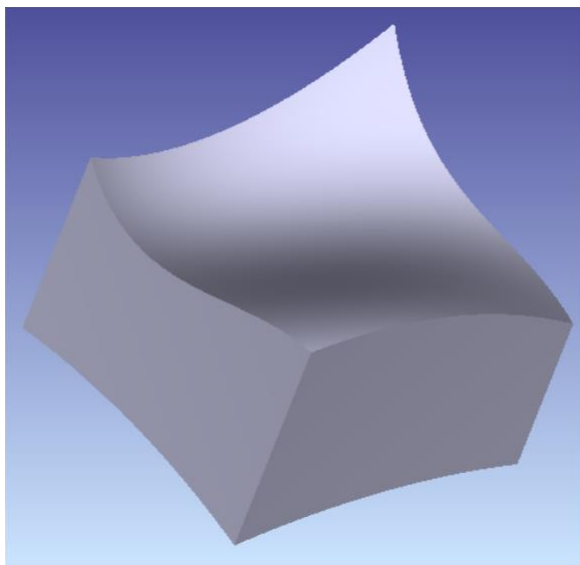
A Biconic Zernike Lens is either rectangular or elliptical in shape, with a standard surface on the front face and a biconic Zernike surface on the rear face. The rear surface is described by the following sag equation:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x{}^2 x^2 - (1 + k_y)c_y{}^2 y^2}} + \sum_{i=1}^{16} \alpha_i x^i + \sum_{i=1}^{16} \beta_i y^i + \sum_{i=1}^{N} A_i Z_i(\rho, \varphi)$$

where

$$c_x = \frac{1}{R_x}, c_y = \frac{1}{R_y}$$

Rx and Ry are the radius of curvature values in the x and y directions, k_x and k_y are the conic constants in the x and y directions, the α terms are the x aspheric coefficients, the β terms are the y aspheric coefficients, N is the number of Zernike coefficients in the series, A_i is the coefficient on the ith Zernike Standard polynomial, *r* is the radial coordinate in lens units, ρ is the normalized radial coordinate, and is the angular coordinate. The Zernike Standard polynomials are defined in the table given in "Zernike Standard Coefficients" .

The following parameters are used to define the Biconic Zernike Lens:

1: The Radial Height of the lens in the y direction in lens units.

2: The X Half-Width in lens units. If this parameter is zero, thenthe outer boundary of the lens will be a circle with a radial size equal to the Radial Height. If this parameter is positive, the outer boundary of the lens will be rectangular. If this parameter is negative, the outer boundary of the lens will be elliptical.

3: The center thickness of the lens.

4, 5: Unused.

6: The front face radius of curvature. Use zero for flat.

7: The front conic constant.

8/9: The rear face radius in the X/Y direction. Use zero for flat.

10/11: The rear face conic in the X/Y direction.

12: The number of Standard Zernike terms.

13: The normalization radius for the Zernike terms.

14-29: The x direction aspheric coefficients.

30-45: The y direction aspheric coefficients.

46 and up: The Zernike Standard coefficients.

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Biconic Zernike Surface

The Biconic Zernike Surface is either rectangular or elliptical in shape, or may have a shape defined by a user defined aperture. The surface is described by the following sag equation:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x^2 x^2 - (1 + k_y)c_y^2 y^2}} + \sum_{i=1}^{16} \alpha_i x^i + \sum_{i=1}^{16} \beta_i y^i + \sum_{i=1}^{N} A_i Z_i(\rho, \varphi)$$

where

$$c_x = \frac{1}{R_x}, c_y = \frac{1}{R_y}$$

Rx and Ry are the radius of curvature values in the x and y directions, k_x and k_y are the conic constants in the x and y directions, the α terms are the x aspheric coefficients, the β terms are the y aspheric coefficients, N is the number of Zernike coefficients in the series, A_i is the coefficient on the ith Zernike Standard polynomial, *r* is the radial coordinate in lens units, ρ is the normalized radial coordinate, and is the angular coordinate. The Zernike Standard polynomials are defined in the table given in "Zernike Standard Coefficients" .

This surface also supports optional X and Y decenters on the Biconic and Zernike terms separately. The biconic decenters are applied to the biconic sag and individual x and y aspheric coefficients, while the Zernike decenters are applied only to the Zernike terms.

The following parameters are used to define the Biconic Zernike Surface:

1/2: The X and Y Half-Width in lens units. If either value is less than zero, the aperture shape will be elliptical, otherwise, the aperture is rectangular, unless a user-defined aperture is applied to the surface.

3/4: The Rx and Ry radii of curvature. Use zero for flat.

5/6: The Kx and Ky conic constants.

7/8: The biconic X and Y direction decenters.

9/10: The Zernike X and Y direction decenters.

11: The number of Standard Zernike terms.

12: The normalization radius for the Zernike terms.

14-29: The x direction aspheric α coefficients.

30-45: The y direction aspheric β coefficients.

46 and up: The Zernike Standard coefficients.

Face Numbers: All faces Face 0.

Binary 1
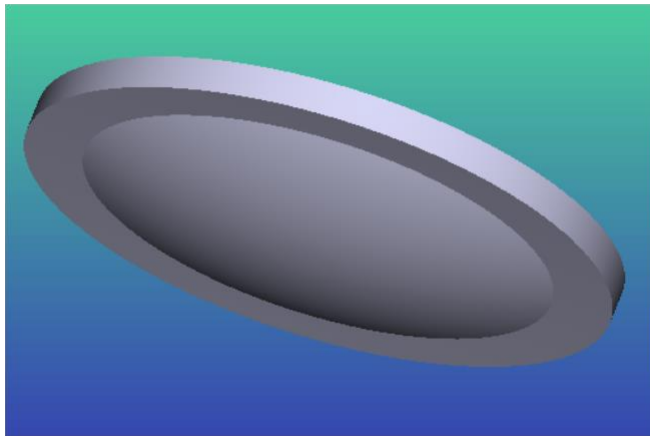
The Binary 1 object is a standard lens with a diffractive optic phase profile similar to the Binary 1 surface type on the front face. The Binary 1 phase profile adds phase to the ray according to the following polynomial expansion:

$$\Phi = M \sum_{i=1}^{N} A_i E_i(x, y)$$

where N is the number of polynomial coefficients in the series, M is the diffraction order, and A_i is the coefficient on the ith extended polynomial term. The polynomials are a power series in the normalized coordinates x and y, as described in "Extended Odd Asphere Lens" . The coefficients A_i units of radians (2π radians is one wave).

The definitions for the parameters are:

1-9: See the Standard Lens description for information on these parameters.

10: The diffraction order M.

11: The normalization radius. The x and y coordinates are normalized to this value. This keeps all coefficients in units of radians.

12: The maximum term number.

13-242: The values of the coefficients on the polynomial terms.

This object does not diffract rays correctly if the front surface is made hyperhemispheric. See also the Binary 2 object.

For important information on diffractive objects, see "Diffraction from NSC objects" .

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.
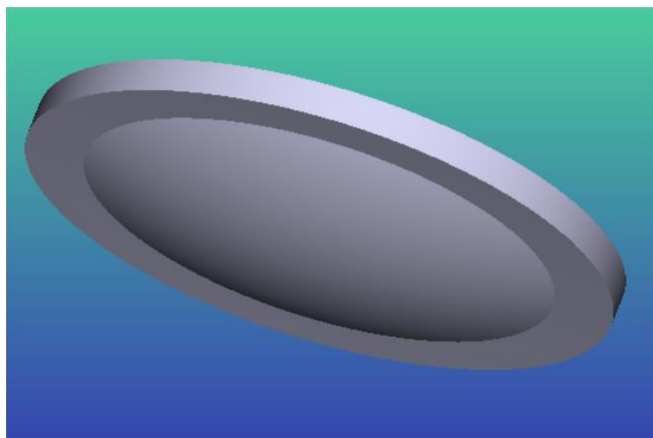
Binary 2

The Binary 2 object is a standard lens with a diffractive optic phase profile similar to the Binary 2 surface type on the front face. The Binary 2 phase profile adds phase to the ray according to the following polynomial expansion:

$$\Phi = M \sum_{i=1}^{N} A_i \rho^{2i}$$

where N is the number of polynomial coefficients in the series, A_i is the coefficient on the $2i^{th}$ power of ρ, which is the normalized radial aperture coordinate. The coefficients A_i all have units of radians (2π radians is one wave).

The definitions for the parameters are:

1-9: See the Standard Lens description for information on these parameters.

10: The diffraction order M.

11: The normalization radius. The x and y coordinates are normalized to this value. This keeps all coefficients in units of radians.

12: The maximum term number.

13-242: The values of the coefficients on the polynomial terms.

This object does not diffract rays correctly if the front surface is made hyperhemispheric. See also the Binary 1 object.

For important information on diffractive objects, see "Diffraction from NSC objects" .

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Binary 2A

The Binary 2A object has an even polynomial aspheric front and back face, with a diffractive optic phase profile similar to the Binary 2 surface type on the front face. The even aspheric sag formula is:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \sum_{i=1}^{12} \alpha_i r^{2i}$$

The Binary 2A object consists of two of these faces, separated by a thickness.

The Binary 2A phase profile is placed on the front face, and adds phase to a ray according to the following polynomial expansion:

$$\Phi = M \sum_{i=1}^{N} A_i \rho^{2i}$$

where N is the number of polynomial coefficients in the series, A_i is the coefficient on the $2i^{th}$ power of ρ , which is the normalized radial aperture coordinate. The coefficients A_i one wave).

The definitions for the parameters are:

1: The Maximum Radial aperture.

2: The Thickness of the lens at the center.

3-4: Unused.

5: The front face radius of curvature.

6: The front face conic constant k.

7-18: The front face coefficients $\alpha 1 - \alpha 12$.

19: The back face radius of curvature.

20: The back face conic constant k.

21-32: The back face coefficients $\alpha 1 - \alpha 12$.

33: The diffraction order M.

34: The normalization radius. The x and y coordinates are normalized to this value. This keeps all coefficients in units of radians.

35: The maximum binary phase term number.

36-235: The values of the coefficients on the polynomial terms.

See also the Binary 1 and Binary 2 objects. For important information on diffractive objects, see "Diffraction from NSC objects" .The reference coordinate is the center of the front face.

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Boolean
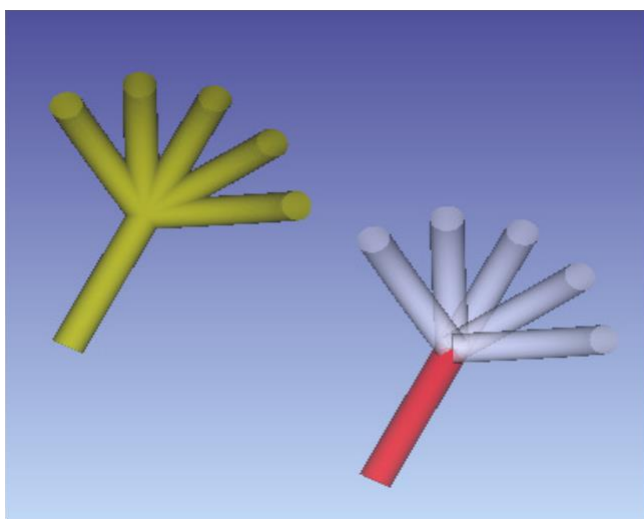
A Boolean object is defined by a series of Boolean operations performed on other objects. Boolean objects can be used to form very general shapes by adding, subtracting, and intersecting other standard shapes. For example, a lens with a hole can be created by defining a Standard Lens object (see "Standard Lens") and then subtracting from the lens a Cylinder Volume (see "Cylinder Volume"). A wide range of complex objects may be created by performing successive Boolean operations on other objects.

The Boolean operations are performed by converting each component object into a NURBS based representation, and then a series of Boolean trimming and combining operations are performed to yield the resulting object. For this reason, Boolean objects have the same considerations that Imported objects do; see "Comments about imported objects" for a discussion. Some loss of precision is possible when converting from Zemax native, high precision

representations to NURBS representations; this is not a limitation of Zemax but is inherent in the NURBS representation of arbitrary surfaces. Generally, precision can be increased as desired using the Spline and Tolerance parameters described below. When extreme optical precision is required, the user should verify that the Boolean object traces to a suitable accuracy.

The Boolean object is defined by 22 parameters and a string defining the desired Boolean operations:

1: Spline controls the number of points to use for fitting splines to aspheric surfaces during the conversion from Zemax format to NURBS format. The value for spline is an integer code, where 0 = 4 points, 1 = 8 points, 2 = 16 points, 3 = 32

points, 4 = 64 points, 5 = 128 points, and 6 = 256 points. Higher spline settings increase accuracy for some objects at the expense of slower object creation, rendering, and ray tracing.

2: Tolerance sets the maximum surface deviation between the actual object and the NURBS representation during creation of the object. Tolerance is an integer with 0, 1, 2 or 3 with the maximum deviation being 1.0E-4, 1.0E-5, 1.0E-6, and 1.0E-7 lens units, respectively. Setting the tolerance integer too high (and the corresponding length tolerance too low) may cause the boolean object to be malformed.

3: Mode: The mode controls the tradeoff between set-up time and ray tracing speed. Use mode 1 for fast set up time and slower ray tracing, mode 2 for medium set up time and medium ray tracing, and mode 3 for slow set up time and fast ray tracing. Generally use mode 1 during set-up of the analysis, and mode 3 for analysis tracing a large number of rays.

4-6: # X Y, Z Voxels: Voxels is a name derived from "volume pixels". A voxel is a 3D rectangle that defines some portion of the total volume occupied by imported solids. Voxel technology allows for fast ray tracing by pre- computing which objects, or portions of objects; lie within a given voxel. A ray entering a voxellated space may only intersect some subset of the total number of voxels; and therefore only these voxels need to be checked for possible ray-object intersections. The greater the number of voxels, the longer the set-up time but the faster the ray tracing. It generally takes some experimenting to determine the optimum number of voxels. Use 5 for all three values if no other values seem obviously superior.

7-12: Unused, reserved for future expansion of this feature.

13-22: The object numbers that define which object is A, B, C, etc. See the discussion below.

Parameters 13-22 are used to define which "parent" objects will be combined. Parameter 13 is used to define which object is Object A, parameter 14 defines which object is Object B, etc. There is no need to define more objects than are required for the desired Boolean operation. If only two objects are to be combined, then only Object A and B need to be defined. Any unused objects may be left with an object number of zero. All defined objects must precede the Boolean object in the NSC Editor. The list of defined objects should always start at Object A, then continue with Object B, etc., e.g. it is not valid to set the object number to zero for Object A and then have a non-zero object number defined later in the list. Parent objects may not be sources or detectors. Once all the parent objects are defined, the Boolean operation is defined by a "Control String" placed in the "Comments" column of the form:

object oper object [oper object]...

The "object" argument is a single letter, such as A, B, or C; which corresponds to the desired object. The "oper" is a single character operator. The following operators (between objects A and B in the logical example) are supported:

+ Combines the two objects together (logical A OR B).

- Subtracts the second object from the first (logical A AND NOT B).

& Computes the intersection of the two objects (logical A AND B).

^ Yields the portion of the object that is part of one or the other, but not both objects (logical A XOR B).

$ Subtracts the first object from the second (logical NOT A AND B). Here are some example Control Strings:

A+B

E+A-B-C

F+B&E-A

All operations are carried out in left-to-right order. No order precedence, such as parentheses, is allowed. Each Boolean operation is performed on the result of previous operations to the left of the operator, with the object listed just to the right of the operator being the second argument. Note the $ operator can be used to subtract the first object from the second. This is useful for combining objects and then subtracting the combination from a third object. For example, to create the object (A - B) and then subtract this combined object from C, use the string "A

- B $ C".

The local coordinate system of the resulting object will be the same as the local coordinate system of the first object listed in the Control String. The global object position and orientation is set by the object position and tilt parameters in the same manner as all NSC objects.

If the parent objects are not needed, use the "Rays Ignore This Object" and "Do Not Draw Object" settings on the parent object (See "Type tab" and "Draw tab"). The Boolean object will still be created even if the parent objects are not drawn or ray traced.

The Face numbers and properties of the Boolean object are defined by the face numbers and properties of the parent objects. All face properties from the parent objects are copied to the corresponding faces of the Boolean object. To view each face on any object, use the NSC Object Viewer (See "NSC Object Viewer"). There is a limit to the total number of unique faces on any single object; for details see "Object faces" .

One significant advantage of the Boolean object is that the resulting volume is still parametrically defined by the parent objects. Changing any of the parent object parameters will change the Boolean object as well. If this capability is not needed, the Boolean object may be exported as a IGES or STEP file (see "Export CAD File") and then imported using the CAD Part: STEP/IGES/SAT object (see "CAD Part: STEP/IGES/SAT"), and then the parent objects may be deleted.

CAD Assembly: Autodesk Inventor , Creo Parametric, SolidWorks
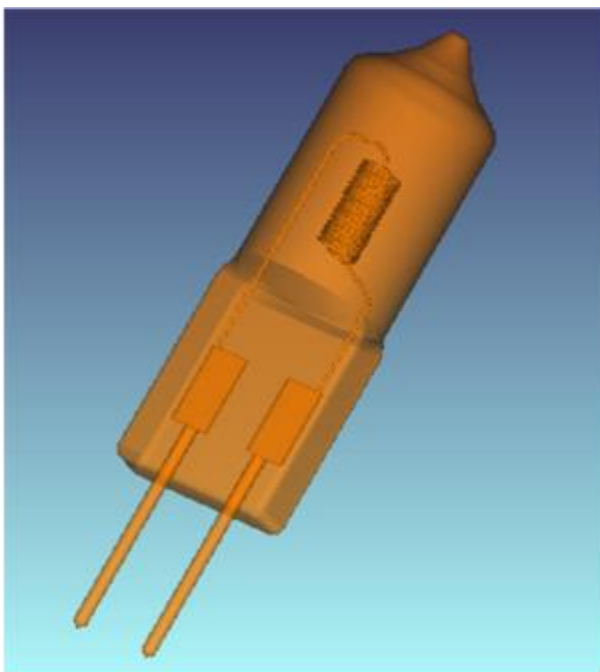
 This feature is only available in the Premium edition of Zemax.

These objects provide a dynamic link to an assembly defined in Autodesk Inventor, Creo Parametric, and SolidWorks, respectively. Autodesk Inventor 2013 or beyond must be installed to use the CAD Assembly: Autodesk Inventor object, which dynamically links to an Autodesk Inventor assembly file (*.iam). Creo Parametric 2.0 or beyond must be installed to use the CAD Assembly: Creo Parametric object, which dynamically links to a Creo Parametric assembly file (*.asm). SolidWorks must be installed to use the CAD Assembly: SolidWorks object, which dynamically links to a SolidWorks assembly file (*.sldasm). All assembly files, along with all of the individual part files (Autodesk Inventor: *.ipt; Creo

Parametric: *.prt; SolidWorks: *.sldprt) that make up each assembly, must be stored in the appropriate folders, which may

be user defined; see "Folders" . Note that the individual part files that make up each assembly are not included when creating an archive of a design that contains an assembly object(see "Backup To Archive File").

By default, the assembly file will be read in as a single object, and thus all of the parts in the assembly may only be assigned a single set of optical properties. These optical properties may already be defined in the file, if the file had been used in a previous Zemax design and the "Save CAD Assembly/

Part Properties" option had been used to save the optical properties in that previous design (see "Save CAD Assembly/Part Properties" under "Edit"). If this is the case, those properties may be read in from the file and assigned to the object in new design. However, this is not mandatory, and new optical properties can be assigned to the file in the new design if desired. The option to read Zemax optical properties from the file will be presented when the file is first loaded, at which point the user may decide the appropriate course of action.

Zemax supports the ability to explode the assembly back into its constituent parts, allowing optical properties to be assigned to each of the individual parts of the assembly. The option to explode the assembly file into individual parts will also be offered when the file is first loaded into Zemax. If this option is not selected upon first load, it may be subsequently selected by choosing either "Explode Autodesk Inventor Assembly", "Explode Creo Parametric Assembly", or "Explode SolidWorks Assembly" (depending on the object selected) under the Tools menu of the Non-Sequential Component Editor. In order for the appropriate tool to be activated, the cursor must be somewhere on the row containing the CAD Assembly object. For more information see "Explode Autodesk Inventor Assembly" , "Explode Creo Parametric Assembly" , or "Explode SolidWorks Assembly" .

When the assembly file is exploded, the constituent parts that are created are each in the form of the CAD Part: STEP/IGES/SAT object (see "CAD Part: STEP/IGES/SAT"). Thus, the individual parts of the assembly file cannot be modified inside of Zemax. If modification of the individual parts is desired, each part should be read into Zemax using the corresponding CAD Part object (see "CAD Part: AutoDesk Inventor" , "CAD Part: Creo Parametric" , and "CAD Part: SolidWorks").

The only parameter associated with the CAD Assembly objects is the "Explode?" flag (parameter 1). Note that manual modification of the "Explode?" parameter does not actually change the behavior of the object. The parameter is there for reference only so Zemax knows when to look for the exploded constituent parts already extracted by the appropriate tool and referenced in the editor. The flag is 0 if the assembly has not been exploded, and is set to 1 if the assembly has been exploded. Once the flag has been set to 1 by the appropriate tool, there is no way to recombine the assembly, other than manually deleting the objects corresponding to the constituent parts. To explode an assembly which has not already been exploded, use the tool described in "Explode Autodesk Inventor Assembly" , "Explode Creo Parametric Assembly" , or "Explode SolidWorks Assembly" .

When using either the CAD Assembly: Creo Parametric object or the CAD Assembly: SolidWorks object, the corresponding application must be opened prior to loading a part into Zemax, and should remain open as long as Zemax is running, to ensure that communication is always available between Zemax and the creating program. Conversely, when using the CAD Assembly: Autodesk Inventor object, it is not necessary to start Autodesk Inventor prior to loading an Inventor part into Zemax. In this latter case, Zemax will automatically open Inventor

in silent mode, meaning that the Inventor application will not appear anywhere on the user's monitor but a process will be visible in the user's Task Manager listing. This process should be allowed to run as long as Zemax is running, in order to ensure that communication is always available between Zemax and Autodesk Inventor.

CAD Part: AutoDesk Inventor, Creo Parametric, SolidWorks
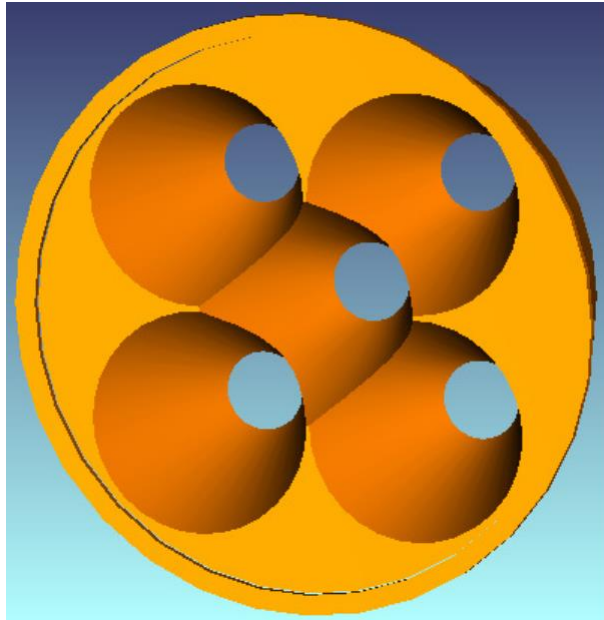
These objects provide a dynamic link to parts defined in Autodesk Inventor, Creo Parametric, and SolidWorks, respectively. Autodesk Inventor 2013 or beyond must be installed to use the CAD Part: Autodesk Inventor object, which dynamically links to an Autodesk Inventor part file (*.ipt). Creo Parametric 2.0 or beyond must be installed to use the CAD Part: Creo Parametric object, which dynamically links to a Creo Parametric part file (*.prt). SolidWorks must be installed to use the CAD Part: Solid- Works object, which dynamically links to a SolidWorks part file (*.sldprt). All part files must be stored in the appropriate folders, which may be user defined; see "Folders" .

Once read in, this object may have optical properties such as coatings, glass, and scattering functions applied, and then be ray traced like any other optical component.

Optical properties for the object may already be defined in the file, if the file had been used in a previous Zemax design and the "Save CAD Assembly/Part Properties" option had been used to save the optical properties in that previous design (see "Save

CAD Assembly/Part Properties" under "Edit"). If this is the case, those properties may be read in from

the file and assigned to the object in new design. However, this is not mandatory, and new optical properties can be assigned to the file in the new design if desired. The option to read Zemax optical properties from the file will be presented when the file is first loaded, at which point the user may decide the appropriate course of action.

Dimensions for the part will be read in by Zemax, and may be displayed directly in the Non-Sequential Component Editor (NSCE) under the parameter columns. The user determines which dimensions to display in the NSCE by selecting those dimensions from either the Autodesk Inventor, Creo Parametric, or SolidWorks Part tab of the Object Properties dialog box, depending on the part selected (see "Autodesk Inventor Part tab" , "Creo Parametric Part Tab" , and "SolidWorks Part tab"). For the CAD Part: Creo Parametric object, the listed dimensions include both global (i.e. model-level) and feature-level parameters defined for the part, i.e. all parameters which are used to define the part geometry analytically.

Zemax will determine the system units used when creating the part in either Autodesk Inventor, Creo Parametric, or SolidWorks ("the creating program"), and will issue a warning if the lens units in Zemax differ from these system units. Zemax will then scale the input part to be the correct size in the given lens units. However, the dimension values associated with the part - i.e. those values that are displayed in the NSCE - will not be scaled, but will always remain in

the system units of the creating program. To avoid potential confusion and error, it is strongly recommended that the user set the Zemax lens units and the system units of the creating program to be the same whenever possible.

The dimension names are taken directly from the creating program, and thus any modification to the names should be done in the creating program. The maximum limit for any dimension name is 50 characters. If this character limit is exceeded, Zemax will issue a warning and will truncate the dimension name to 50 characters.

No dimensions will be displayed when the part is first loaded into Zemax. Any dimensions which are subsequently chosen for display in the NSCE (via the Part tab for the creating program in the Object Properties dialog box) may be modified in Zemax, analogous to modification of a built-in Zemax object. The modified part

may then be saved into a file with the format of the creating program (i.e. *.ipt, *.prt, or *.sldprt) if desired. The saved file may be given the same name as the original part (thus overwriting the original file) or may be given a new name.

Dimension values associated with the part can be defined as optimization variables, may be placed under multi-configuration control, and can be perturbed or used as compensators during tolerance analysis, just like any other parameter of a built-in Zemax object. The position and tilt of the part may also be optimized and toleranced in the usual manner.

When using either the CAD Part: Creo Parametric object or the CAD Part: SolidWorks object, the correspond- ing application must be opened prior to loading a part into Zemax, and should remain open as long as Zemax is running, to ensure that communication is always available between Zemax and the creating program. Conversely, when using the CAD Part: Autodesk Inventor object, it is not necessary to start Autodesk Inventor prior to loading an Inventor part into Zemax. In this latter case, Zemax will automatically open Inventor in silent mode, meaning that the Inventor application will not appear anywhere on the user's monitor but a process will be visible in the user's Task Manager listing. This process should be allowed to run as long as Zemax is running, in order to ensure that communication is always available between Zemax and Autodesk Inventor. In all cases, constant communi- cation between Zemax and the creating program is necessary to allow modification of the part inside of Zemax.

CAD Part: STEP/IGES/SAT

Objects may be imported in either IGES, SAT, or STEP format. These file formats are commonly supported by CAD programs. Imported objects may also be in ZOF (Zemax's native format, see "Imported objects and ZOF files"). Once imported, an object may have optical properties such as coatings, glass, and scattering functions applied, and then be ray traced like any other optical component.

To import an object, set the object type to "Imported" and select the file name from the drop-down box, or place the file name in the comment column. The file must reside in the <objects>\CAD Files folder (see "Folders") and end in the extension IGS, IGES, SAT, STP, STEP, or ZOF. The extension must be included with the file name on the comment column.

The following parameters are used by the Import object type.

1: Scale: This dimensionless parameter scales the entire imported solid. Upon import, Zemax will automatically attempt to scale the dimensions of the imported solid to match the current dimensions in Zemax; this scale factor is applied after that conversion.

2: Mode: The mode controls the trade off between set-up time and ray tracing speed. Use mode 1 for fast set up time and slower ray tracing, mode 2 for medium set up time and medium ray tracing, and mode 3 for slow set up time and fast ray tracing. Generally use mode 1 during set-up of the analysis, and mode 3 for analysis tracing a large number of rays.

3-5: # X Y, Z Voxels: Voxels is a name derived from "volume pixels". A voxel is a 3D rectangle that defines some portion of the total volume occupied by imported solids. Voxel technology allows for fast ray tracing by pre- computing which objects, or portions of objects; lie within a given voxel. A ray entering a voxellated space may only intersect some subset of the total number of voxels; and therefore only these voxels need to be checked for possible ray-object

intersections. The greater the number of voxels, the longer the set-up time but the faster the ray tracing. It generally takes some experimenting to determine the optimum number of voxels. Use 5 for all three values if no other values seem obviously superior.

6 - 8: Unused.

9: Explode?: This parameter indicates whether the object has been exploded into its constituent parts (for objects that represent assemblies). Note that manual modification of the "Explode?" parameter does not actually change the behavior of the object. The parameter is there for reference only so Zemax knows when to look for the exploded constituent parts already extracted by the appropriate tool and referenced in the editor. The flag is 0 if the object has not been exploded, and is set to 1 if the object has been exploded. Once the flag has been set to 1 by the appropriate tool, there is no way to recombine the object, other than manually deleting the objects corresponding to the constituent parts. To explode an imported object which has not already been exploded, use the tool described in "Explode CAD Part: STEP/IGES/SAT" .

Imported objects and ZOF files

When importing CAD objects from IGES, STEP, or other format files, the data must be translated into the Zemax internal object representation. This conversion need only be done once, and thereafter the converted file can be used. This greatly speeds up the loading of imported objects. The converted file name is the original file name with the extension ZOF (Zemax Object Format) appended. For example, if the original imported file was MyObject.IGS, the ZOF file created will be MyObject.IGS.ZOF. The name of the file used by the imported object will be changed to the converted file name. Zemax will automatically convert imported files to ZOF format files if the original file requires more than 2.0 seconds to read and convert, and if the scale factor is 1.0. If the scale factor needs to be something other than 1.0, first set the scale factor to 1.0, then after the file has been converted to ZOF format, the scale factor may be adjusted to the desired value. This restriction prevents unexpected scaling of the ZOF object when more than one ZMX file is using the same ZOF object.

It is recommended that the original CAD file be retained, even though the ZOF file will be used by Zemax. This allows regeneration of the ZOF file should that be required. This feature may be disabled, see "Convert Imported Files To ZOF" under "Type tab" .

Comments about imported objects

The advantage to using imported objects is that solids of any shape may be ray traced within Zemax. There are no limits on the shape, complexity, or number of objects that may be imported and ray traced. Multiple objects may be imported in a single file. The disadvantage to using imported objects is ray tracing speed, and in some cases, ray tracing accuracy. See the discussion below for important considerations regarding imported objects. An alternative to imported objects exists, see "User Defined Object" .

Ray tracing accuracy for imported objects

Not all types of surface shapes may be ray traced with adequate accuracy using representations supported by CAD file formats, such as IGES, SAT, and STEP. For planes, spheres, and cylinders, the CAD representation, if done correctly, is of very high precision suitable for optical accuracy ray tracing. However, higher order shapes do not usually have a native representation in CAD formats. For example, an aspheric surface with a polynomial term of the form r^16 may have no equivalent representation in the chosen CAD format. A CAD program will generally approximate this shape using a segmented spline, which is in general a piece-wise fit of the surface using multiple lower order polynomials. Typically, multiple third or fourth order polynomials are used to approxi- mate the surface. This is probably adequate for mechanical design, but not for optical precision ray tracing, where surfaces must be know to tiny fractions of the wavelength of light.

This problem often arises when a high optical precision surface is modeled in Zemax, then exported as a CAD file, then imported as an CAD file for subsequent ray tracing. The optical precision of the part is lost upon exporting the native Zemax asphere as a CAD spline. See "Limitations of exported data" for more information.

For non-imaging optics, the precision of the CAD representation is usually adequate, but for imaging systems, great care must be taken to verify that the imported CAD part is a suitably accurate description of the desired shape. Note Zemax

uses a relative internal optical precision of about 1E-12 for ray tracing. Most CAD representations of objects are many orders of magnitude more coarse.

Ray tracing speed for imported objects

Simple objects such as spherical lenses typically ray trace slower when imported in CAD format than the native Zemax object of identical shape. Ray tracing speed for imported objects is critically dependent upon the efficient representation of the solid shape within the imported file. The identical object may be represented in a nearly infinite number of ways using the various solid and surface entity types supported by the CAD formats Zemax can import. For example, an efficient representation of an object may use only a few spline surfaces; while an inefficient representation of the object may use hundreds of smaller spline surfaces. Although from a mechanical modeling perspective the two representations may both be valid and the resulting solids identical, the represen- tation with the larger number of spline surfaces will ray trace dramatically slower. The only remedy is to return to the source of the CAD file and see if a more efficient representation may be generated.

Limitations on imported objects

Not all possible valid CAD format files may be imported, as Zemax may only import solids. No lines or surfaces are allowed. Shells must be converted to thin solids before being imported into Zemax. Solids must be simply closed with a continuous exterior surface without holes or gaps. No internal surfaces or faces are allowed. Solids composed of multiple solid volumes either in contact or overlapping cannot be ray traced. Multiple volumes not in contact are allowed. Files containing valid solids which do not import may be sent to technical support for review; however no guarantee is offered that Zemax can be made to accommodate all possible CAD format entity types and files.

CAD Part: STL
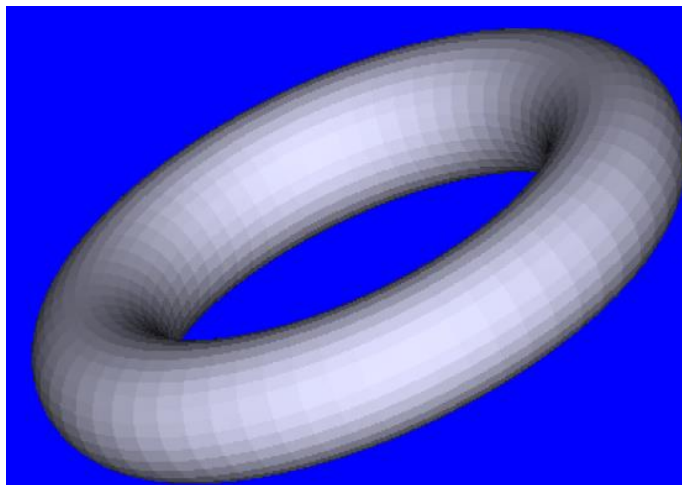
The STL object is a very general user-defined object. It can be used to define an open polygon surface or a closed polygon volume such a prism or other solid. The STL Object format is based upon a collection of 3D triangles. This format is widely supported by mechanical CAD programs. Both the text and the binary variations of the STL file format are supported. See the section "Defining STL objects" which follows for details. See also "User Defined Object" , "Polygon Object" , and "CAD Part: STEP/IGES/SAT" .

There are no fixed limits to the total number of vertices or polygons.

The STL file name is referenced in the "comments" column of the STL Object row. For example, if the STL file myobject.STL is placed in the <objects>\CAD Files folder (see "Folders"), then specify "myobject.STL" in the comment column of the STL Object type row in the NSC Editor.

The reference coordinate is locally (0, 0, 0), and the polygons that compose the object may be placed anywhere relative to the reference point. Some STL export implementations only allow objects to be placed such that all vertex coordinates are positive. Zemax does not require this to be the case, and will import triangle vertices anywhere in 3D space.

Using an STL object as a detector is supported, see "Objects as detectors" . When the STL object is used as a detector, If "Fast Ray Trace" (see "Type tab") is checked, Zemax will alter the order of the triangles defined in the STL file to speed up the ray trace. If "Fast Ray Trace" is unchecked, the order of the triangles is unmodified, which slows the ray tracing but preserves the original order of the triangles defined in the STL file.

STL objects require two parameters:

1: A scale factor. All vertices in the STL file are multiplied by this parameter.

2: A flag to indicate if the STL file defines a volume or a surface. If the "Is Volume?" parameter is zero, then Zemax assumes the STL file defines an open surface. Rays must either reflect or absorb from such a surface; refraction is not allowed. If the "Is Volume?" parameter is any non-zero value, then Zemax assumes the STL file defines a closed volume. The volume may be reflecting, refracting, or absorbing.

Face Numbers: All faces Face 0.

Defining STL Objects

The STL format is commonly supported by mechanical CAD programs to describe arbitrary objects. Objects are modeled as a collection of triangles, and the global vertices of the triangles are written out to a file. STL is a good format for faceted objects. For smoothly curved objects, such as lenses, STL is an approximation which may be of acceptable accuracy for some non-imaging systems.

There are binary and text formats for STL files; and Zemax supports both formats.

To use an STL object, choose the object type to be an "STL Object" and enter the name of the file (without the STL extension) in the comments column of the STL Object row. The STL file must be placed in the folder <objects>\CAD Files (see "Folders").

Maximum triangles in STL objects

There is no fixed upper limit on the number of triangles an STL object may contain. The ultimate limit is determined by the amount of real or virtual RAM the computer has available. Each triangle takes up about 100 bytes of memory. However, Zemax often maintains multiple copies of the lens data simultaneously, and so a good rule of thumb is that Zemax will need 500 bytes of RAM for each triangle. A 2000 triangle object will require about

1 Mb of free RAM. A more practical limit is computer speed; Zemax will slow down noticeably if the number of triangles becomes very large.

Example STL files

Several example STL files may be found in the <objects>\CAD Files folder (see "Folders").

CAD Part: Zemax Part Designer
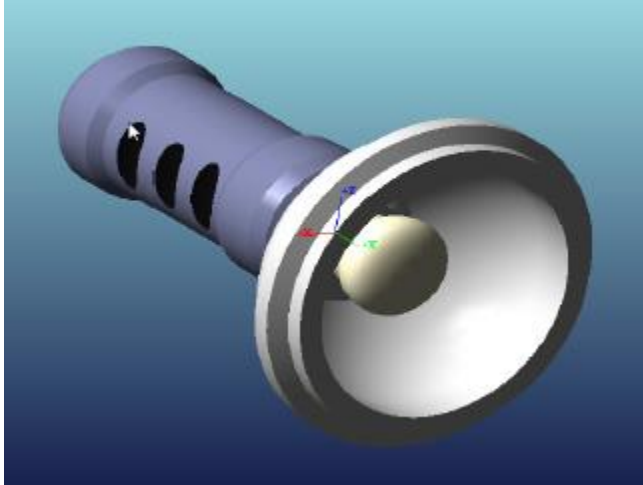
 This feature is only available in the Premium edition of Zemax.

A Part Designer Object is a solid volume created using the Zemax Part Designer. A wide variety of complex parametric parts may be created using the Part Designer. For a complete description of Part Designer see the program documentation for the Part Designer.

Compound Parabolic Concentrator (CPC)
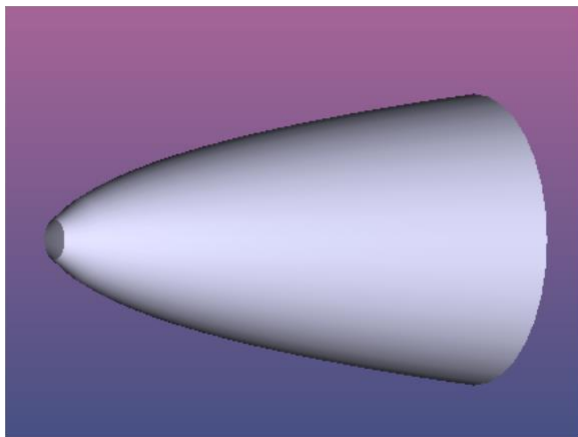
A CPC is defined by the following parameters:

1: The radial aperture at z = 0, a.

2: The maximum acceptance angle in degrees, θ .

3: The length along the local Z axis, L.

4-5: Unused.

6: The "Is Volume?" flag.

7: The volume index of refraction.

A CPC is used to concentrate light entering one end of the CPC to the other end. Only rays that make an angle less than the acceptance angle with respect to the local Z axis will pass through the CPC; other rays will reflect back out. This type of CPC is the "Basic CPC" as described in detail in "High Collection Nonimaging Optics" by W. T. Welford and R Winston, Academic Press (1989).

If the "Is Volume?" flag is zero, then the object is a hollow shell. Otherwise, the object is a closed solid volume. If the CPC is a closed volume, it may be made of a refractive material. The reference index of the material can be entered for parameter 7. This index will be used to scale the acceptance angle (via Snell's law) and the refracted acceptance angle will be used to determine the CPC shape and maximum length.

The maximum value for the CPC length is given by:

$$L = \frac{a(1 + \sin\theta)}{\tan\theta\sin\theta}$$

Longer lengths will be truncated to this value. The radial coordinate of points on the CPC as a function of the z coordinate along the axis is given by the positive real root of this quadratic equation:

$$C^2 r^2 + 2(CSz + aP^2)r + (z^2 S^2 - 2aCQz - a^2 PT) = 0$$

where

$$C = \cos\theta,\ S = \sin\theta,\ P = 1 + S,\ Q = 1 + P,\ \text{and}\ T = 1 + Q$$

The acceptance angle must be between 0.0 and 89.0 degrees, inclusive. Face Numbers: Front face Face 1, rear face Face 2, outer faces Face 0

CPC Rectangular
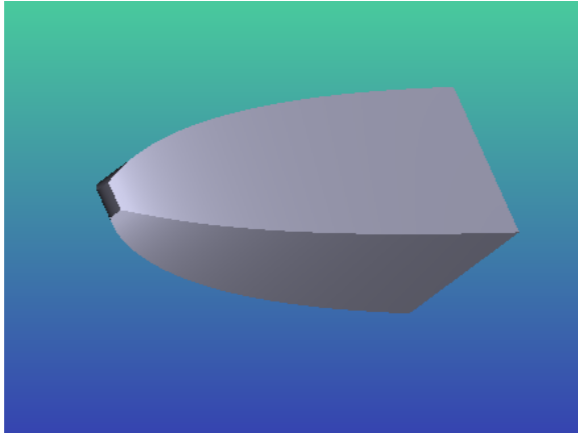
This object is similar to the CPC (see "Compound Parabolic Concentrator (CPC)"), but with rectangular rather than radial symmetry. See that object description for detailed equations defining the CPC.

The CPC Rectangular is defined by the following parameters:

1: The x direction semi-aperture at z = 0.

2: The maximum x-direction acceptance angle in degrees.

3: The y direction semi-aperture at z = 0.

4: The maximum y-direction acceptance angle in degrees.

5: The length along the local Z axis.

6: The "Is Volume?" flag.

7: The volume index of refraction.

The acceptance angles must be between 0.0 and 89.0 degrees, inclusive.

Face Numbers: Front face Face 1, rear face Face 2, outer faces Face 0.

Cone
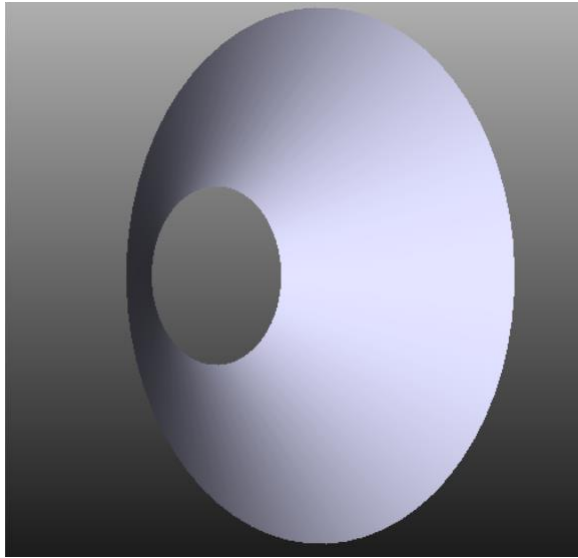
A cone is defined by 4 parameters:

1: The z coordinate of the first point.

2: The radial coordinate of the first point.

3: The z coordinate of the second point.

4: The radial coordinate of the second point.

The line segment defined by the points is rotated about the z axis to form a section of a cone. This object can be used to make an annular or circular shape (if the two z coordinates are identical) or a cylindrical shape (if the two r coordinates are identical). In this sense the cone is redundant with the annulus and cylinder pipe objects. The cone is used as a primitive for creating Fresnel lenses.

The reference coordinate is locally (0, 0, 0), and the points that define the cone may be placed anywhere relative to the referencepoint. Face Numbers: All faces Face 0.

Cylinder Pipe
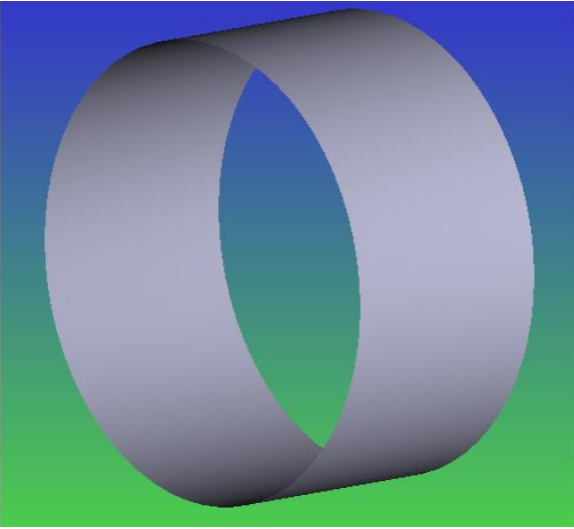
A cylinder pipe is a rotationally symmetric surface defined b parameters:

1: The radius of the front circular aperture.

2: The length along the local Z axis of the cylinder.

3: The radius of the rear circular aperture.

This object is normally used to define a reflective light pipe. The reference coordinate is the center of the front aperture

Face Numbers: All faces Face 0

Cylinder Volume
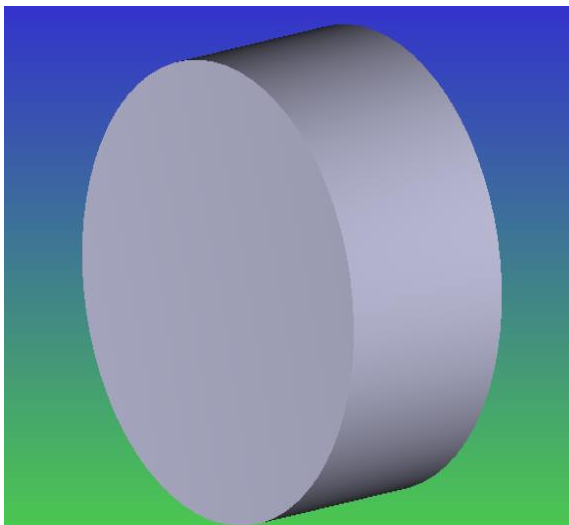
A cylinder volume is a rotationally symmetric volume defined by 3 parameters:

1: The radius of the front circular face.

2: The length along the local Z axis of the cylinder.

3: The radius of the rear circular face.

This object is very similar to the cylinder pipe, except the front and rear faces are included to make the shape a closed volume. Because the object is a closed volume, it may be reflective, refractive, or absorbing.

The reference coordinate is the center of the front aperture. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Cylinder 2 Pipe
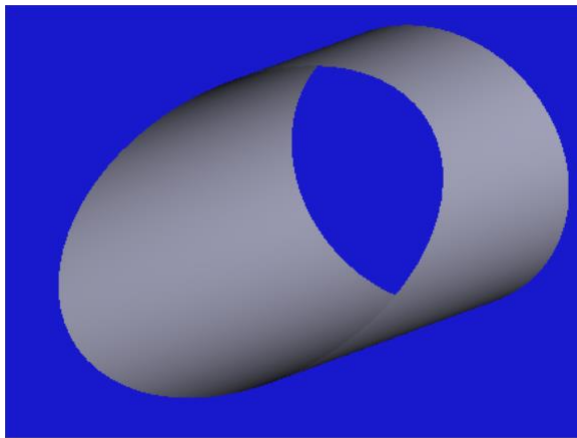
A cylinder 2 pipe is a rotationally symmetric surface defined by 4 parameters:

1: The radius of the cylinder.

2: The length along the local Z axis of the cylinder.

3: The tilt of the front face along the y direction in degrees.

4: The tilt of the rear face along the y direction in degrees.

5: The tilt of the front face along the x direction in degrees.

6: The tilt of the rear face along the x direction in degrees.

The reference coordinate is the center of the front face.

Face Numbers: All faces Face 0.

Cylinder 2 Volume
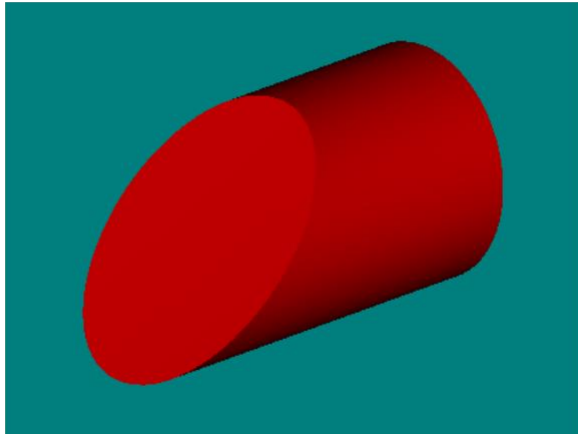
A cylinder volume is a rotationally symmetric volume defined by 4 parameters:

1: The radius of the cylinder.

2: The length along the local Z axis of the cylinder.

3: The tilt of the front face along the y direction in degrees.

4: The tilt of the rear face along the y direction in degrees.

5: The tilt of the front face along the x direction in degrees.

6: The tilt of the rear face along the x direction in degrees.

This object is very similar to the cylinder 2 pipe, except the front and rear faces are included to make the shape a closed volume.

Because the object is a closed volume, it may be reflective, refractive, or absorbing.

The reference coordinate is the center of the front aperture.

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Diffraction Grating
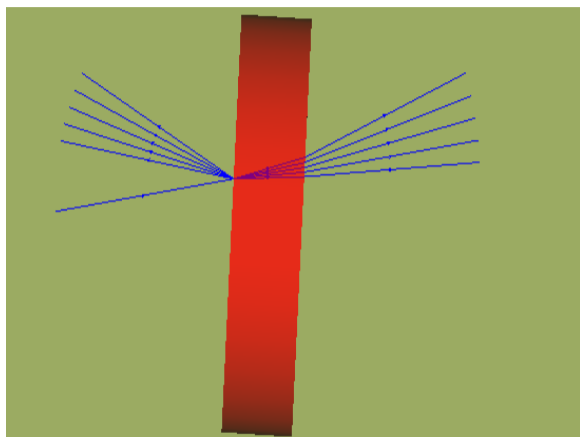
The diffraction grating is a Standard Lens with a diffraction grating on one face. For a rectangular diffraction grating, see "Lenslet Array 1" . The Diffraction Grating is defined by the following parameters:

1-9: See "Standard Lens" for information on these parameters.

10: The grating line frequency in lines/micrometer on the front face. See parameter 12 below.

11: The diffraction order for the front face.

12: The grating formula. If zero, the grating is a constant spacing linear grating with a grating frequency defined by parameter 10. If the formula is "1", the grating frequency in lines/micrometer is defined by

$$ t = t_0 + t_1 y + t_2 y^2 + t_3 y^3 + t_4 y^4 $$

where t_0 is defined by parameter 10 and the other coefficients are defined by parameters 13-16.

The grating is assumed to consist of lines parallel to the local x axis. The grating frequency is the lines per micrometer along the y direction; projected down on to the surface.

This object does not diffract rays correctly if the front diffractive surface is made hyperhemispheric. For important information on diffractive objects, see "Diffraction from NSC objects" .

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Dual BEF Surface

The Dual Brightness Enhancement Film (BEF) Surface is an idealized surface that can reflect and transmit rays with variable amplitude depending upon the polarization state of the ray. The surface is always a rectangular plane, and may not have any material, coatings, or scattering function applied. The surface may be embedded in any isotropic, homogeneous medium. If either ray splitting or polarization is off, the surface has no optical effect. Otherwise, any incident ray is split into a transmitted ray and a reflected ray. The transmission and reflection intensity coefficients are

defined as parameters 3-6, and are different for X and Y polarized light, where the X and Y directions are defined in the surface coordinate system. The transmission and reflection for X polarized light may not exceed 1, and the same limit applies to Y polarized light.

Note that both reflected and transmitted electric fields must remain orthogonal to the respective ray vectors, and so for some combinations of incident angle, polarization state, and coefficient values, energy in the transmitted and/or reflected rays may not be conserved. If the transmitted or reflected energy is greater than the incident energy, Zemax will scale the energy to the incident intensity. In rare cases the sum of the transmitted and reflected energy may not obey energy conservation. The 6 parameters used to define this object are:

1-2: The X and Y Half Widths.

3-6: Reflection X, Reflection Y, Transmission X, Transmission Y

Face Numbers: All faces Face 0

Ellipse
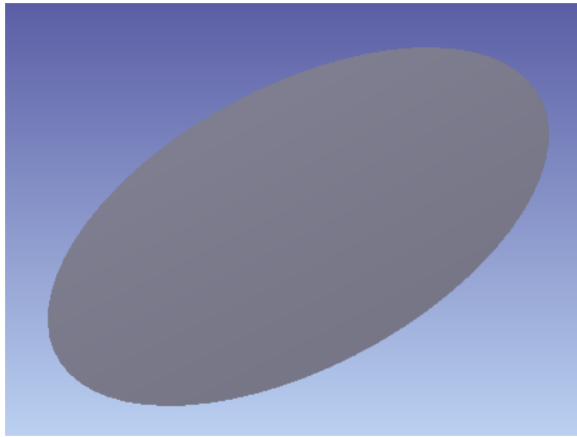
The ellipse is a plane elliptical surface shape defined by 2 parameters:

1: The X Maximum Half Width.

2: The Y Maximum Half Width.

The ellipse resides entirely within the local XY plane. This object is a special case of the more general annulus. The reference coordinate is the center of the ellipse.

Face Numbers: All faces Face 0.

Elliptical Volume
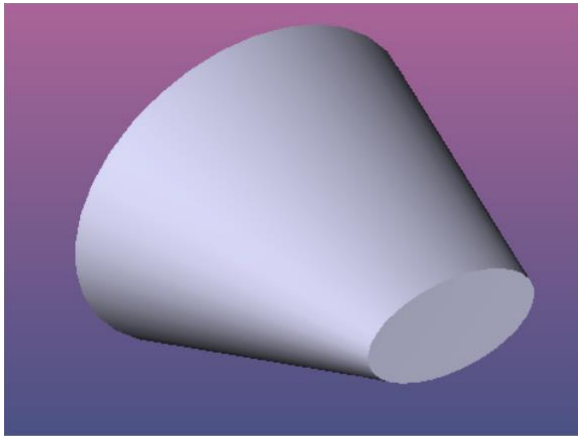
An elliptical volume is a tapered volume or surface with an elliptical cross section defined by these parameters:

1: The x half width of the front face.

2. The y half width of the front face.

3: The x half width of the back face.

4. The y half width of the back face.

5: The length along the local Z axis of the cylinder.

6-7: Unused.

8: The "Is Volume?" flag. Use 1 to make a closed volume, else

use 0 to make a hollow shell.

This object is similar to the cylinder pipe and volume. Note this

object may be either a hollow shell or a closed volume, depending upon the Is Volume? flag.

The reference coordinate is the center of the front face.

Face Numbers: Front face Face 1, back face Face 2,all other faces Face 0.

Even Asphere Lens
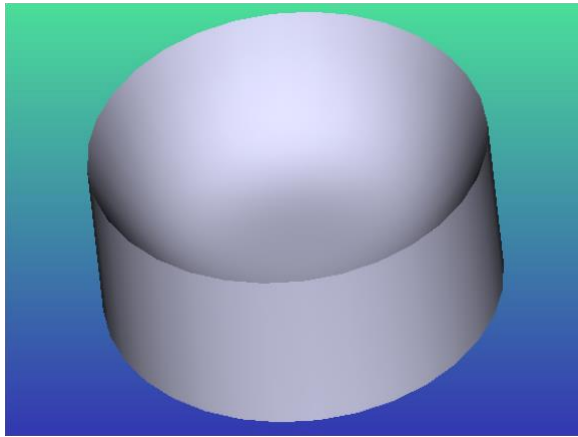
The Even Asphere surface shape is defined by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}$$

which is exactly the same sag formula as for the Even Asphere surface. The Even Asphere Lens object consists of two of these faces, separated by a thickness. The total object shape is defined by 24 parameters:

1: The Maximum Radial aperture.

2: The Thickness of the lens at the center.

3-4: Unused.

5: The front face radius of curvature.

6: The front face conic constant k.

7-14: The front face coefficients $\alpha_1 - \alpha_8$.

15: The back face radius of curvature.

16: The back face conic constant k.

17-24: The back face coefficients $\alpha_1 - \alpha_8$.

The reference coordinate is the center of the front face.

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Extended Odd Asphere Lens
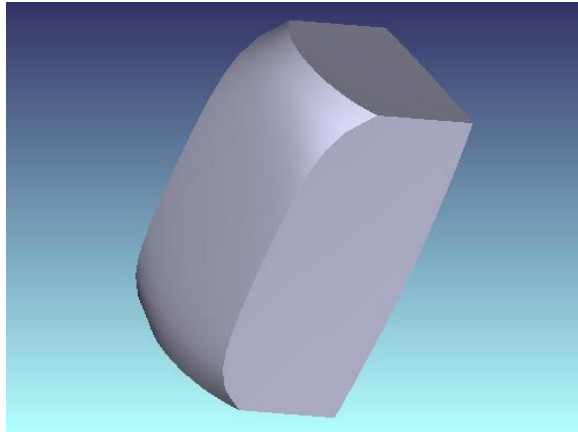
The Extended Odd asphere surface shape is defined by the following equation:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \sum_{i=1}^{N} \alpha_i \rho^i$$

where N is the number of polynomial coefficients in the series and α_i is the coefficient on the ith extended polynomial term. The normalized radial coordinate ρ is used so that the coefficients α_i all have units of lens units. The maximum order number on the polynomial terms is 118 for each face.

The Extended Odd Asphere Lens object is composed of two of these faces, separated by a thickness. The object shape is defined by the following parameters:

1: The radial height of the lens object in lens units. This value is used for the y direction half height if the lens is rectangular or elliptical.

2: The x half-width in lens units. If this parameter is zero, then the outer boundary of the lens will be a circle with a radial size equal to the Radial Height. If this parameter is positive, the outer boundary of the lens will be rectangular. If this parameter is nega- tive, the outer boundary of the lens will be elliptical.

3: The center thickness of the lens in lens units.

4: The front face radius of curvature. If this value is zero, then the curvature is assumed to be zero.

5: The front face conic constant k.

6: The front face normalization radius.

7: The front face number of extended polynomial terms.

8: The rear face radius of curvature. If this value is zero, then the curvature is assumed to be zero.

9: The rear face conic constant k.

10: The rear face normalization radius.

11: The rear face number of extended polynomial terms.

12 and on: The front and rear face extended polynomial coefficients. The rear face coefficients follow the front face coefficients.

The reference coordinate is the center of the front face.

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Extended Polynomial Lens
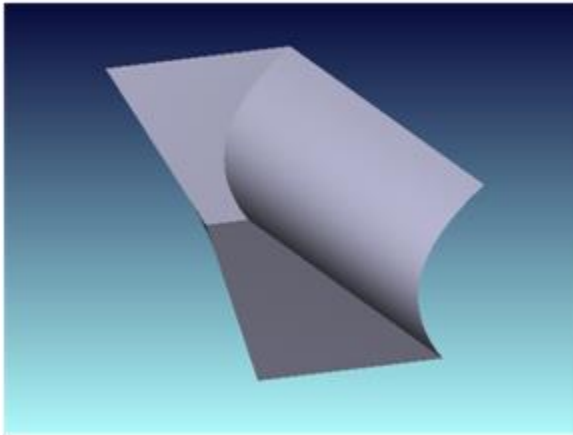
The Extended Polynomial surface shape is defined by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \sum_{i=1}^{N} A_i E_i(x, y)$$

where N is the number of polynomial coefficients in the series, and A_i is the coefficient on the ith extended polynomial term. The polynomials are a power series in x and y. The first term is x, then y, then x*x, x*y, y*y, etc. There are 2 terms of order 1, 3 terms of order 2, 4 terms of order 3, etc. The maximum order is 20, which makes a maximum of 230 polynomial aspheric coefficients. The position values x and y are divided by a normalization radius so the polynomial coefficients are dimensionless.

The Extended Polynomial Lens object consists of two of these faces, separated by a thickness. The total object shape is defined by these parameters:

1: The radial height of the lens object in lens units. This value is used for the y direction half height if the lens is rectangular.

2: The x half width of the lens object in lens units. If zero, the lens is circular; otherwise, the lens is rectangular.

3: The center thickness of the lens in lens units.

4-5: Unused.

6: The front face radius of curvature. If this value is zero, then the curvature is assumed to be zero.

7: The front face conic constant k.

8: The front face normalization radius.

9: The front face number of extended polynomial terms.

10: The rear face radius of curvature. If this value is zero, then the curvature is assumed to be zero.

11: The rear face conic constant k.

12: The rear face normalization radius.

13: The rear face number of extended polynomial terms.

14 and on: The front and rear face extended polynomial coefficients. The rear face coefficients follow the front face coefficients.

The reference coordinate is the center of the front face. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Extended Polynomial Surface
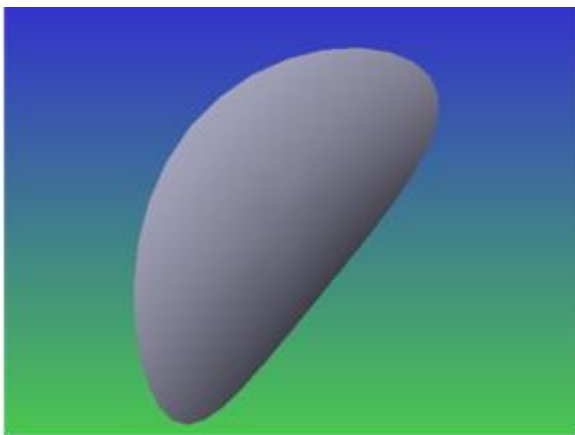
This object is very similar to the Extended Polynomial Lens object. Rather than being a solid, the object is a shell surface with only 1 set of extended asphere coefficients. The object is defined by these parameters:

1: The radial height of the surface. This object supports user defined aperture shapes, see "User defined apertures" .

2: The radius of curvature.

3: The conic constant k.

4-5: Unused.

6: The normalization radius.

7: The number of extended polynomial terms.

8 and on: The extended polynomial coefficients.

The reference coordinate is the center of the surface. Face Numbers: All faces Face 0.

Extruded
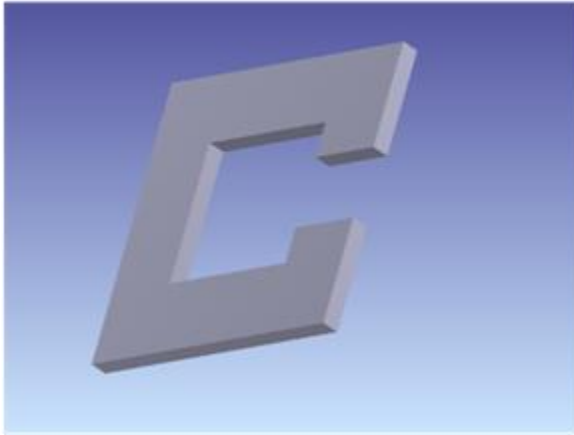
An Extruded object is a User Defined Aperture extruded to form a solid volume. Any User Defined Aperture file may be used to create the Extruded object. For information on User Defined Apertures see "User defined apertures and obscurations" .

The extrusion is along the Z axis, and the front and rear faces of the object may be independently and anamorphically scaled. The rear face may also be decentered relative to the front face.

The User Defined Aperture file name is placed in the comment column. The following additional parameters are used:

1: The length along the Z axis between the front and rear faces.

2-3: The front face X and Y direction scaling factors.

4-5: The rear face X and Y direction scaling factors.

6-7: The rear face X and Y direction decenters.

The reference coordinate is the origin of the front face. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Faceted Surface
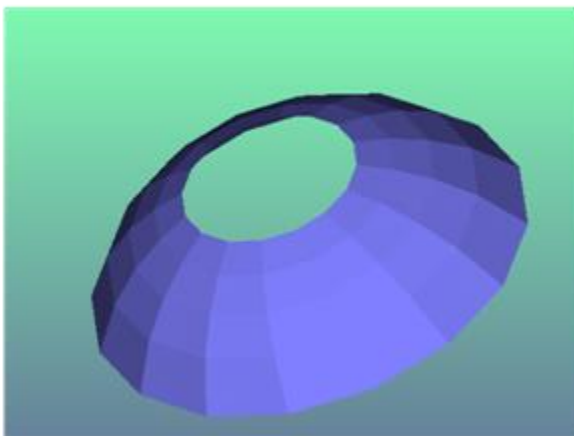
A Faceted Surface is a standard surface shape faceted into flat polygons. The following additional parameters are used:

1: The radius of curvature.

2: The conic constant.

3: The shape; use 0 for annular, 1 for elliptical, 2 for rectangular, and 3 for toroidal.

4-5: For shape 0, these are the minimum and maximum radial apertures; otherwise these are the X/Y half widths.

6-7: The number of facets to use in making the surface into polygons.

The reference coordinate is the origin of the front face. All faces Face 0.

Freeform Z

A Freeform Z object is formed by drawing a cubic spline curve through a series of points in the YZ plane, and then sweeping this curve around the Z axis to form either a solid volume or shell. The Freeform Z object is defined by two parameters and a user selectable number of defining points:

1: The number of points. This is the number of coordinate pairs used to define the curve in the YZ plane. The minimum is 5, the maximum is 124.

2: A flag to indicate if the resulting object is a solid volume or a hollow shell. If the "Is Volume?" parameter is zero, then Zemax assumes the shape is hollow shell. No "end caps" are added in this case. If the object is a solid volume, then flat "end caps" are added if the first or last points do not have a y coordinate of zero to close the solid.

3-250: The coordinate values of the YZ curve, in pairs.

The Z coordinate of the first point must be zero. No Z value may be less than or equal to the Z value of the preceding point. The absolute value of the Y values is always taken when creating the spline curve.

There is a special optimization operand designed to provide boundary constraints on the shape of Freeform Z objects, see "FREZ" for more information.

A tool is available that greatly simplifies inserting and deleting points in the spline curve; see "Insert/Delete Z Point Tools" .

The reference point is the center of the front face of the lens. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Fresnel 1
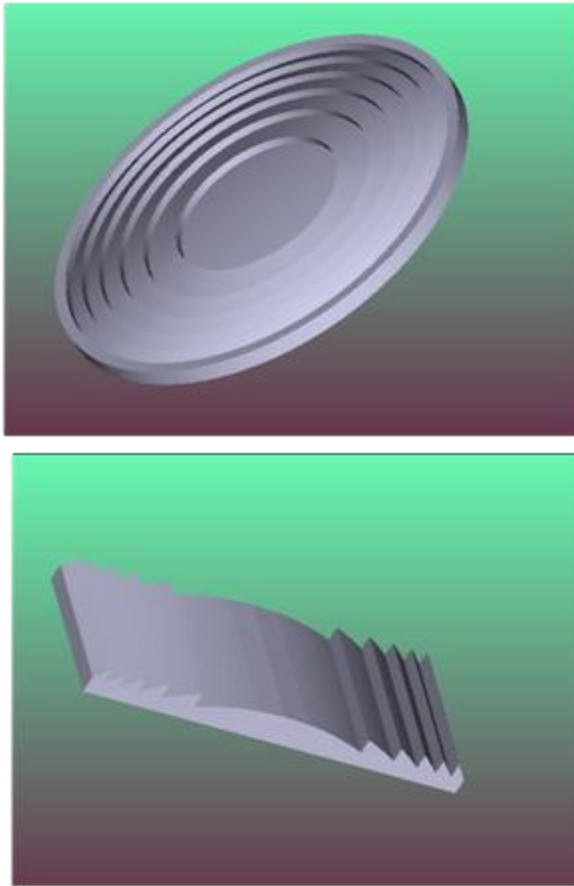
This object is a general radially symmetric or cylindrical solid Fresnel lens constructed by modeling the detailed faces of the Fresnel surface. For an idealized Fresnel lens (which ignores the detailed structure of the Fresnel surface and thus ray traces much faster) see the Fresnel 2 object.

The substrate shape is a flat disk (if radial) or a rectangle (if cylindrical). One face of the substrate consists of radial or rectangular faces defining the profile of the Fresnel that yields optical power. The profile is constructed of radially flat faces (or a series of flat faces if subsegments are used) whose endpoints are defined by a sag expression identical to the Even Asphere surface:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}$$

However, to make a Fresnel lens, each face is offset just enough along the z axis so all faces start at the same z coordinate as the center vertex point. This yields a lens that is "collapsed" into a relatively small volume. The Fresnel faces are automatically generated by Zemax using these 16 parameters:

1: Radial Height: This is the maximum radial aperture of the lens if radially symmetric, or the y half height if cylindrically symmetric.

2: X Half Width: This is the half width of the lens if cylindrically symmetric. If this parameter is zero, then a rotationally symmetric lens is generated.

3: +Depth/-Frequency: If this parameter is positive, then it corresponds to the depth of each groove in lens units. If negative, then it corresponds to the frequency of the grooves. For example, a value of -2.0 will yield 2 grooves per radial lens unit. If the groove depth is defined, the radial positions of the grooves will generally vary; if the groove frequency is defined; the groove depth will vary. For the case of a defined groove depth Zemax automatically computes the exact radial coordinate at which the sag has changed by the specified depth. This is done using an iterative search.

4: Pitch (degrees): The pitch is the angle the "inactive" faces (those faces nominal parallel to the local z axis) make with respect to the z axis. The pitch is always radially outward, whether the pitch angle is positive or negative. A pitch angle of a few degrees is typically added to Fresnel molds to make extraction of the molded part easier.

5: Thick: The thickness of the Fresnel in lens units. This value may be positive or negative; but should be chosen such that the absolute value of the thickness exceeds the deepest groove depth; or a non-physical Fresnel may be generated without warning or error message.

6: Radius: The base radius of curvature. This is one over the value "c" in the sag expression above.

7: Conic: The conic constant "k" in the sag expression above.

8-15: The coefficients on the even radial powers of r. Note these coefficients have units as r is not normalized.

16: The number of sub segments. The greater the number of sub segments, the smoother the approximation to a curved surface between the grooves. A value of 1 yields flat grooves, larger values yield progressively more smooth curved faces at the expense of ray tracing speed. The maximum number of sub segments is 20.

Because the object is a closed volume, it may be reflective, refractive, or absorbing.

The reference coordinate is the center vertex of the side of the lens with the Fresnel faces. If the radius or aspheric terms yield negative sag values, then it is important to offset the position of the Fresnel such that the entire solid resides inside the non-sequential group. If the entry port is placed inside the lens; incorrect ray tracing will result.

Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Fresnel 2
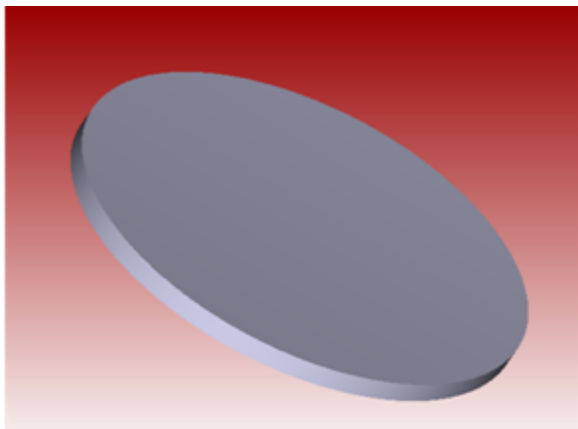
This object is an idealized Fresnel lens. Unlike the Fresnel 1, this Fresnel lens uses the approximation that the Fresnel faces are infinitesimally small, and may be ignored for purposes of computing the ray-object intersection point. Unlike the idealized Fresnel surface in sequential ray tracing (see "Fresnel"), this object does not add or subtract any phase to the ray.

The substrate shape is either a flat disk or a rectangle. The front face of the substrate consists of a radial or cylindrical Fresnel lens. The radial profile is defined by a sag expression identical to the Even Asphere surface:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}$$

If the Fresnel surface is cylindrical, then the profile is described by an identical expression in the y coordinate alone:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14} + \alpha_8 y^{16}$$

The Fresnel lens may be either radial or cylindrical independent of the substrate shape. It is possible to define a radial Fresnel on a rectangular substrate or a cylindrical Fresnel on a radial substrate if desired.

The Fresnel 2 object is defined using these 14 parameters:

1: Radial Height: This is the maximum radial aperture of the lens if radially symmetric, or the y half height if cylindrically symmetric.

2: X Half Width: This is the half width of the lens if cylindrically symmetric. If this parameter is zero, then a rotationally symmetric lens is generated.

3: Thick: The thickness of the Fresnel in lens units. This value must be positive.

4: Is Cylinder: If zero, the Fresnel surface will be radial, otherwise, the surface will be cylindrical.

5: Radius: The base radius of curvature. This is one over the value "c" in the sag expressions above.

6: Conic: The conic constant "k" in the sag expressions above.

7-14: The coefficients on the even radial powers of r if "Is Cylinder" is zero, otherwise these are the coefficients on the even radial powers of y. Note these coefficients have units as y and r are not normalized.

Because the object is a closed volume, it may be reflective, refractive, or absorbing.

The reference coordinate is the center vertex of the side of the lens with the Fresnel faces. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Grid Sag Lens
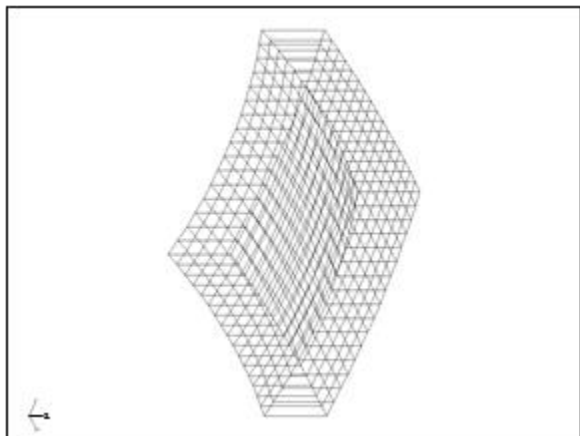
The Grid Sag Lens is a solid volume object with a Standard surface front face, and a Grid Sag surface rear face. The Standard surface shape is described in "Standard" . The Grid Sag surface shape is described in "Grid Sag" .

Grid Sag Lens objects use the following parameters:

1: The x half width in lens units. See the discussion below.

2: The y half width in lens units. See the discussion below.

3: The front surface radius of curvature in lens units.

4: The front surface conic in lens units.

5: The axial lens thickness in lens units.

6: Interpolate: Use 0 for bicubic, and 1 for linear. See "Bicubic spline vs. linear interpolation" .

7: The back surface radius of curvature in lens units.

8: The back surface conic constant.

9-16: The back surface α_i aspheric coefficients.

17-18: The X- and Y-decenter values in lens units. This is the decenter of the grid data relative to the aspheric substrate. The decenter values are added to any decenter values defined in the GRD file, described below.

If the x and y half widths are zero, the lens shape is rectangular and the size is defined by the data in the GRD file. If the x half width is zero, and the absolute value of the y-half width is not zero but is less than the size of the data in the GRD file, then the part is square (if the y half width is greater than zero) or circular (if the y half width is less than zero). If both the x half width and the y half width are not zero, and the absolute values of both half widths are less than the size of the data in the GRD file, then the part is rectangular (if both half widths are positive) or elliptical (if either half width is negative).

The actual point by point sag data is defined by placing the data in a properly formatted file ending in the extension GRD. The file format is defined in "Importing grid data" . The grid data file must be placed

in the folder <objects>\Grid Files (see "Folders"). The file name, without the path, is then placed in the comment column of the NSC Editor.

The reference coordinate is the center of the front face. Face Numbers: Side faces Face 0, front face Face 1, back face Face 2.

Grid Sag Surface
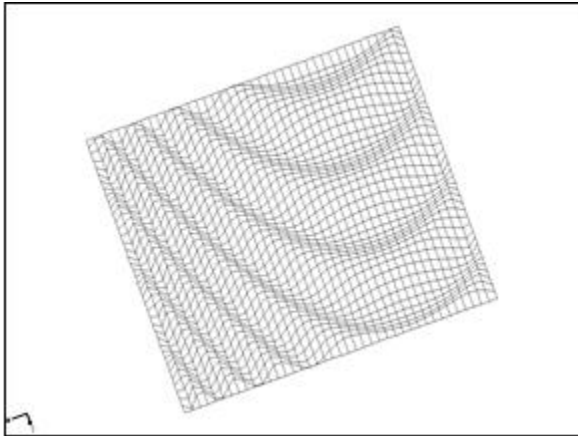
The Grid Sag Surface is an object whose shape is defined by a rectangular array of points listed in a file. This surface shape is similar to the sequential Grid Sag surface, although no Zernike terms are supported. For a detailed description of the sequential Grid Sag surface shape, see "Grid Sag" .

Grid Sag Surface objects use the following parameters:

1: Interpolate: Use 0 for bicubic, and 1 for linear. See "Bicubic spline vs. linear interpolation" .

2: The radius of curvature in lens units.

3: The conic constant.

4-11: The α_i aspheric coefficients.

12-13: The X- and Y-decenter values in lens units. This is the decenter of the grid data relative to the aspheric substrate. The decenter values are added to any decenter values defined in the GRD file, described below.

The actual point by point sag data is defined by placing the data in a properly formatted file ending in the extension GRD. The file format is defined in "Importing grid data" . The grid data file must be placed in the <objects>\Grid Files folder (see "Folders"). The file name, without the path, is then placed in the comment column of the NSC Editor.

Face Numbers: The entire face is Face 0.

Hexagonal Lenslet Array
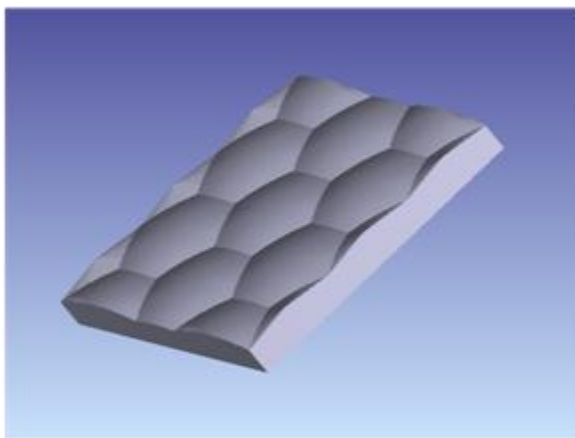
The hexagonal lenslet array object models a rectangular volume with a front face consisting of a hexagonal array of even aspheric lenslets. To make the part rectangular, partial hexagonal lenses are added along the sides as required. The even asphere surface sag expression is identical to that of the Even Asphere Lens object (see "Even Asphere Lens"). The back face is plano. The object uses the following parameters:

1: The number of columns (must be an odd positive integer).

2: The number of rows (must be an odd positive integer).

3: Draw as flat: If this is set to 1, the front face of the lenslet array is drawn as flat. Zemax uses this flat face as a first guess for raytracing and then iterates to the actual curved array surface. This can speed up rendering times, but shouldn't be used if rays have large angles of incidence on the front face of the object.

4:Width: This is the full width of a single hexagon lenslet in lens units.

5: Thickness: The thickness of the lenslet, measured along the +z axis from the center lenslet vertex to the back plano face.

6: Radius: The radius of curvature of each lenslet in lens units.

7: Conic: The conic constant of each lenslet.

8-15: The aspheric coefficients on the powers of r.

The reference coordinate is the center of the front face. Face Numbers: Side faces Face 0, front face Face 1, back face Face 2.

Hologram Lens
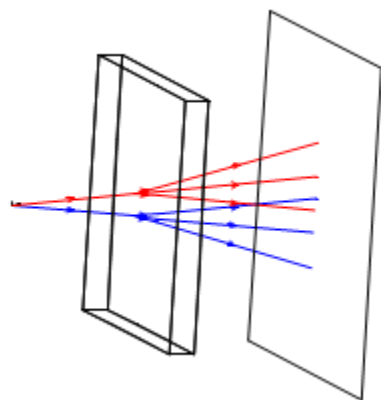
This object is an ideal optically fabricated hologram similar to the Hologram 1 and Hologram 2 sequential surface models (see "Hologram 1" for a description of these surfaces). The hologram is a solid, and may be circular or rectangular in shape. The front and back faces may be plane, spheres, or conic aspheres. The hologram surface is on the front face. See also "Hologram Surface" .

The Hologram object is defined using these 18 parameters:

1: The radial height of the lens object in lens units. This value is used for the y direction half height if the lens is rectangular.

2: The x half width of the lens object in lens units. If zero, the lens is circular; otherwise, the lens is rectangular.

3: The center thickness of the lens in lens units.

4-5: Unused.

6: Radius 1: The radius of the front face.

7: Conic 1: The conic constant of the front face.

8: Radius 2: The radius of the back face.

9: Conic 2: The conic constant of the back face.

10: Holo Type: Use 1 for Hologram type 1 (both sources converging/diverging) or 2 for Hologram type 2 (one source converging, one diverging) See "Hologram 1" .

11: Order: The diffraction order to use. Multiple orders may be specified, see "Diffraction tab" .

12: Construction Wavelength: The wavelength in micrometers used to fabricate the hologram.

13-18: The X, Y, and Z coordinates in lens units of the construction points relative to the vertex of the front face of the Hologram.

Because the object is a closed volume, it may be air, reflective, refractive, or absorbing. If the order is zero or if the ray totally internally reflects at the hologram boundary, no hologram diffraction is computed.

For important information on diffractive objects, see "Diffraction from NSC objects" .

The reference coordinate is the center of the front face. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Hologram Surface
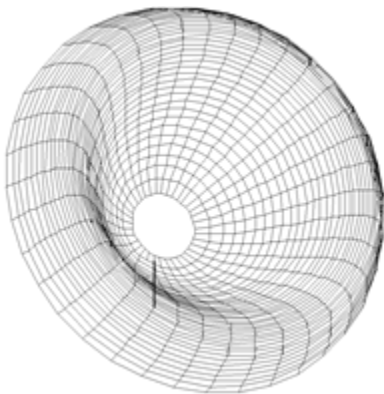
This object is an ideal optically fabricated hologram similar to the Hologram 1 and Hologram 2 sequential surface models (see "Hologram 1" for a description of these surfaces). The hologram is a surface, and may be circular or user defined in shape. The surface shape may be plane, sphere, or a conic and/or polynomial asphere. See also "Hologram Lens" .

The surface shape is defined by the following sag equation:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + \sum_{i=1}^{N} \alpha_i r^i$$

where c is the curvature of the surface, k is the conic constant, r is the radial coordinate, and the α terms are aspheric coefficients. The surface supports specification of both a minimum and a maximum radial aperture; so annular surfaces may be defined. Note that both even and odd terms are defined; up to approximately 230 coefficients may be used. This is the same shape as the Aspheric Surface object, see "Aspheric Surface" .

The following parameters are used to define the hologram surface:

1: The radius of curvature. If this value is zero, then the curva- ture is assumed to be zero.

2: The conic constant k.

3: The maximum radial aperture in lens units.

4: The minimum radial aperture in lens units. This value may be zero.

5-6: Unused.

7: Holo Type: Use 1 for Hologram type 1 (both sources converg- ing/diverging) or 2 for Hologram type 2 (one source converging, one diverging) See "Hologram 1" .

8: Order: The diffraction order to use. Multiple orders may be specified, see "Diffraction tab" .

9: Construction Wavelength: The wavelength in micrometers used to fabricate the hologram.

10-15: The X, Y, and Z coordinates in lens units of the construction points relative to the vertex of the front face of the Hologram.

16: The number of terms to use in the aspheric expansion. Ray tracing will be faster if this term is no larger than the highest order non-zero coefficient.

17-250: The α coefficients on the polynomial expansion.

Face Numbers: All faces Face 0.This object supports user defined apertures, see "User defined apertures" .

Jones Matrix

The Jones Matrix is a plane elliptical surface whose shape is defined by 2 parameters, and polarization transmission/reflection properties by 8 "ABCD" parameters:

1: The X Maximum Half Width.

2: The Y Maximum Half Width.

3-10: Ar, Ai, Br, Bi, Cr, Ci, Dr, Di

The ellipse resides entirely within the local XY plane.

The Jones Matrix may be used to model neutral density filters, polarizers, and rotators. For a description of the Jones Matrix parameters, see "Defining polarizing components" .

The reference coordinate is the center of the ellipse. Face Numbers: All faces Face 0. This object supports user defined apertures, see "User defined apertures" .

Lenslet Array 1
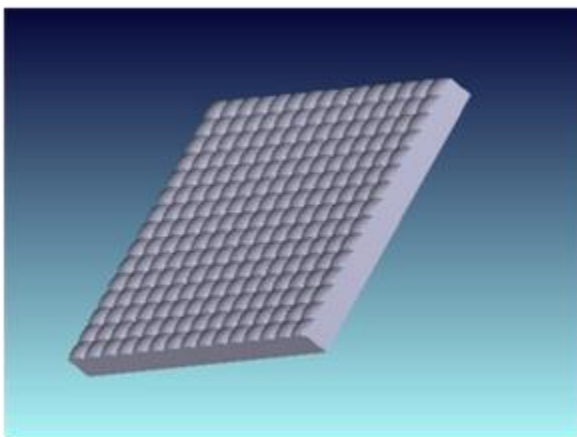
 The array aspects of this object are provided for back compatibility with older versions of Zemax. A more efficient method that ray-traces more quickly is to make a single lenslet and then create an array using the Array Object - see "Array" for details.

A Lenslet Array 1 object consists of an array of rectangular volumes, each with a flat front face and a curved back face. The back face may be plane, sphere, conic, or polynomial asphere; or a spherical, conic, or polynomial aspheric toroid. The shape may be decentered with respect to the center of the lenslet. A toroidal surface is defined by a curve in the YZ plane which is then rotated about an axis parallel to the Y axis but displaced by a distance R; called the radius of rotation.

If the back face is rotationally symmetric, the radial profile is defined by a sag expression identical to the Even Asphere surface:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}$$

If the back face is a toric, then the profile is described by an identical expression in the y coordinate alone:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14} + \alpha_8 y^{16}$$

where c is the reciprocal of the radius of curvature in YZ plane.

Either the front or the back surface may be a diffraction grating. The grating is assumed to consist of equally spaced lines parallel to the local x axis. The grating frequency is the lines per micrometer along the y direction; projected down on to the surface.

The Lenslet Array 1 is defined by these parameters:

1: The X Half-Width in lens units of each lenslet.

2: The Y Half-Width in lens units of each lenslet.

3: The thickness along the local Z axis of each lenslet.

4-5: Unused.

6: The radius of curvature of each lenslet, use zero for a plane.

7: The conic constant of each lenslet.

8: The "Is Toric" flag; if zero the lenslet surface is rotationally symmetric, otherwise it is toroidal.

9: The radius of rotation of each lenslet if the lenslet surface is toric.

10: The grating line frequency in lines/micrometer on either the front or back face, see parameter 24.

11: The diffraction order. Multiple orders may be specified, see "Diffraction tab" .

12-19: The aspheric coefficients 1-8.

20-21: The decenter x and decenter y of the curved back face.

22-23: The number of lenslet elements in x and y. If the total number of lenslets is large, it is generally much faster to use a single lenslet (set parameters 22 and 23 both equal to 1) and then use the Array object to create the array of lenslets. The Array object method is much more powerful, uses less memory, and will generally load and trace much faster than using a large number of lenslet elements. See "Array" .

24: The face to make diffractive. This value must be 1 (front face) or 2 (back face).

Note a cylinder lens results if the radius of rotation is set to zero. The reference coordinate is the center of the front face. Face Numbers: Outside edge side faces are Face 0, flat front face Face 1, curved back face Face 2, inside side faces Face 3 (these are the side faces formed in the interior of an array of lenslets when decentered lens faces meet).

Lenslet Array 2
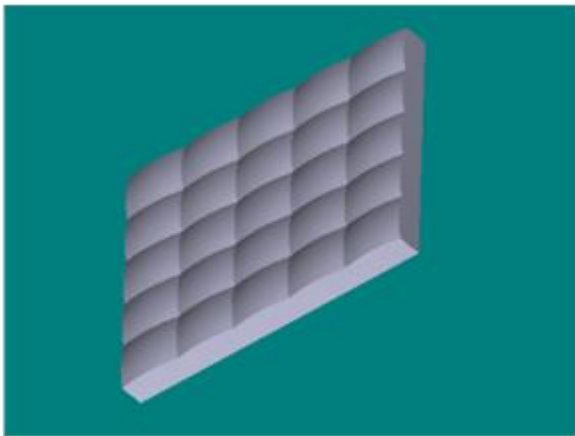
 The array aspects of this object are provided for back compatibility with older versions of Zemax. A more efficient method that ray-traces more quickly is to make a single lenslet and then create an array using the Array Object - see "Array" for details.

A Lenslet Array 2 object consists of an array of rectangular volumes, each with curved front and back faces. The faces may be plane, sphere, or conic asphere.

The Lenslet Array 2 is defined by these parameters:

1: The X Half-Width in lens units of each lenslet.

2: The Y Half-Width in lens units of each lenslet.

3: The thickness along the local Z axis of each lenslet.

4-5: Unused.

6: The radius of curvature of the front face of each lenslet, use zero for a plane.

7: The conic constant of the front face of each lenslet.

8: The radius of curvature of the back face of each lenslet, use zero for a plane.

9: The conic constant of the back face of each lenslet.

10-11: The number of lenslet elements in x and y. If the total number of lenslets is large, it is generally much faster to use a single lenslet (set parameters 10 and 11 both equal to 1) and then use the Array object to create the array of lenslets. The Array object method is much more powerful, uses less memory, and will generally load and trace much faster than using a large number of lenslet elements. See "Array" .

The reference coordinate is the center of the front face. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Micro Electro Mechanical System (MEMS)
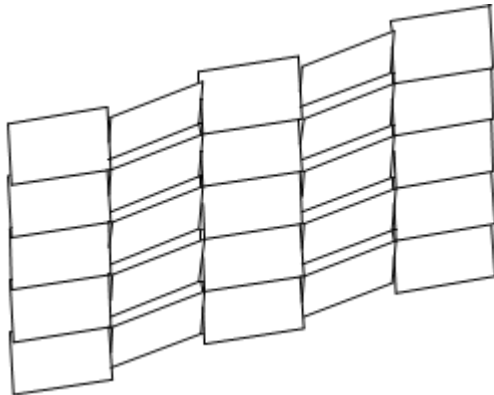
This object simulates a Micro Electro Mechanical System (MEMS). The MEMS consists of an array of small rectangular mirrors (commonly called pixels). The mirrors may tip at any of three angles, each rotated about an axis to point the mirror in any direction. The mirrors may be turned on and off by rows, by columns, or by individual mirrors if desired to model any state the MEMS can be in. This type of device is sometimes called a Digital Mirror Device (DMD). The following parameters are used to define the MEMS:

1: The number of X pixels across each row.

2: The number of Y pixels down each column.

3: The full X width of the array in lens units. The X pixel width is this number divided by the number of X pixels.

4: The full Y width of the array in lens units. The Y pixel width is this number divided by the number of Y pixels.

5, 6, 7: The tip angle of the pixel when in state 0, 1, or 2.

8: The rotation angle around the Z axis about which the pixel will tip. This angle is measured clockwise from the +Y axis.

9: P-Flag. If this value is 0, then the pixels are addressed by rows. If 1, then the pixels are addressed by columns, and if 2, the pixels are addressed individually.

10 and above: Integer values which define the state of the rows, columns, or pixels, as described below.

The pixels are set to either state 0, 1, or 2 using a series of base-3 integer values. To determine the values for any logic state of the MEMS, construct a table similar to the one below, which to save space just shows the values for 3 rows or columns or pixels (r/c/p):

r/c/p:

3 2 1 X

0 0 0 0

0 0 1 1

0 0 2 2

0 1 0 3

0 1 1 4

0 1 2 5

0 2 0 6

etc..

where the value X is entered into parameter 10 to control the first 15 r/c/p; parameter 11 to control r/c/p 16-30, and so on for as many r/c/p exist. Because of the way Zemax addresses the individual r/c/p; the total number of addressable pixels cannot exceed approximately 3750. This limits the number of x pixels, y pixels, or the product of x and y pixels, depending upon the address mode defined by the P-flag (parameter 9).

Note that X for any given state of the MEMS is given by: $X = M1*(3^0) + M2*(3^1) + M3*(3^2) + ...$

where M1 is the logic state (0, 1, or 2) of the first row/column/pixel and M2 is the logic state of the second row/column/pixel, etc. Up to 15 r/c/p values are defined by each parameter value in the editor. The pixels are numbered starting in the bottom left corner with pixel #1, and incrementing across the columns and then up along the rows.

Face Numbers: All faces Face 0.

Null Object

This is a non-existent object. It can be used as a place holder or a reference point for other objects. The reference coordinate is locally (0, 0, 0).

Odd Asphere Lens
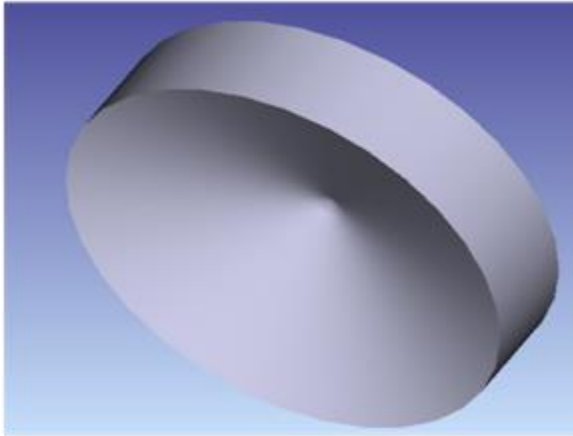
The Odd Asphere surface shape is defined by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2 r^2}} + \sum_{i=1}^{12} \alpha_i r^i$$

which is very similar to the Odd Asphere surface (there are 4 additional terms). The Odd Asphere Lens object consists of two of these faces, separated by a thickness. The total object shape is defined by 32 parameters:

1: The Maximum Radial aperture.

2: The Thickness of the lens at the center.

3-4: Unused.

5: The front face radius of curvature.

6: The front face conic constant.

7-18: The front face coefficients $\alpha 1 - \alpha 12$.

19: The back face radius of curvature.

20: The back face conic constant.

21-32: The back face coefficients $\alpha 1 - \alpha 12$.

The reference coordinate is the center of the front face. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Paraxial Lens

The Paraxial Lens is an ideal thin lens surface with separate X and Y focal lengths. The defining parameters are:

1/2: The X/Y Half Widths. Make positive for a rectangular shape, negative for an elliptical shape.

3/4: The X/Y direction focal length, measured in air, in lens units.

This object may be placed in air, or be fully or partially immersed within a homogeneous or gradient index medium. No adjustment of the optical path length or phase of the ray is made during refraction, so no attempt at estimating the correct phase of the ray is made. No attempt is made to compute the polarization effects of refraction through this idealized part. If polarization effects are important, this object should be used with caution, especially if the ray angles on either side of the lens are large with respect to the normal.

Polygon Object

The polygon object is a very general user-defined object. It can be used to define an open polygon surface or a closed polygon volume with some portions reflective and others refractive or absorptive. The Polygon Object is based on a collection of 3D triangles whose vertices are placed in a file with the POB extension. See the "Defining Polygon Objects" for more details. Any Polygon Object may be used as a detector as described in "Objects as detectors" .

There are no fixed limits to the total number of vertices or polygons. The POB file name is referenced in the "comments" column of the Polygon Object row. For example, if the POB file myobject.POB is placed in the <objects>\Polygon Objects folder (see "Folders"), then specify "myobject.POB" in the comment column of the Polygon Object type row in the NSC Editor. Polygon objects require two parameters:

1: A scale factor. All vertices in the POB file are multiplied by this parameter.

2: A flag to indicate if the POB file defines a volume or a surface. If the "Is Volume?" parameter is zero, then Zemax assumes the POB file defines an open surface. If the "Is Volume?" parameter is any non-zero value, then Zemax assumes the POB file defines a closed volume.

See "Special considerations for faceted objects" for information on limitations of ray tracing through faceted objects. A tool for creating POB files for use as detectors is described in "Create Polygon Object" .

The reference coordinate is locally (0, 0, 0), and the polygons that compose the object may be placed anywhere relative to the reference point. Face Numbers: Each face has a face number as defined in the POB file, see below for details.

Defining Polygon Objects

Polygon objects are user-defined objects that are composed of a collection of triangles in 3D space. The triangles are defined in a text file ending in the extension POB (for polygon object). Each face, or a group of faces, on the polygon may be assigned a unique face number if desired. There is a limit to the total number of unique faces on any single object; for details see "Object faces" .

Polygon objects that form closed volumes must not contain any interior faces. Every face must lie on the exterior of the object, defining the boundary between the inside and the outside of the volume. To make an object that has a cemented boundary, such as a beam splitter, use two or more separate polygon objects and then place them in contact. For more information on placing objects in contact, see "Placing objects inside of, adjacent to, or overlapping other objects".

The file format may be created and edited with any text editor. The file consists of a series of rows of data. Each row begins with a single letter or symbol, followed by data for that symbol. The supported symbols and their meanings are defined as follows.

The comment symbol: !

The symbol "!" is used to define a comment line.

Syntax:

! Any comment here

Example:

! A dove prism

The face name symbol: C

Names for the faces may be defined using the C command.

Syntax:

C facenumber "any name here"

Example:

C 0 "All faces"

The face names are used to display in the object dialog box the names for the various faces of an object. Note the name must be in quotes. This command is optional.

The invisible symbol: I

Lines which connect vertices that are not to be drawn (such as lines between two adjacent triangles that lie in the same plane) can be marked as invisible. To mark two vertices as invisible, use the symbol "I" followed by the two vertex numbers. Any line connecting the marked vertices will not be drawn.

This only affects the 3D Layout plot, and is strictly a cosmetic enhancement to the drawing of polygon objects.

Syntax:

I v1 v2

Example:

I 7 9

Note the vertex numbers must be integers. The numbers are separated by spaces. The I command must precede any triangles or rectangles that reference the invisible vertices.

The vertex symbol: V

Vertices are defined by the symbol "V" followed by the vertex number and the x, y, and z coordinates of the vertex.

Syntax:

V number x y z

Example:

V 1 -1.0 -1.0 0.0

Note the vertex number must be an integer, and the x, y, and z coordinates are floating point numbers. The numbers are separated by spaces.

The triangle symbol: T

Triangles are defined by the connection of 3 vertices.

Syntax:

T vertex1 vertex2 vertex3 isreflective face

The vertex numbers must be integer vertices previously defined in the file. The "isreflective" flag is -1 if the surface absorbs, 1 if the surface reflects, or 0 if the surface refracts. Using this flag allows some triangles to be reflective, and others to be refractive or absorptive, within the same Polygon Object or face. However, the isreflective flag is not considered in the object produced by Boolean operations with the Polygon Object.

The value for face defines which face the triangle belongs to. If the face number is omitted; face 0 is assumed.

Example:

T 1 2 3 0 2

This would define a refractive triangle connecting vertices 1, 2, and 3, belonging to face 2.

The rectangle symbol: R

Rectangles are defined by the connection of 4 vertices, otherwise, they are very similar to triangles. Internally, Zemax converts rectangles into 2 triangles.

Syntax:

R vertex1 vertex2 vertex3 vertex4 isreflective face

The vertex numbers must be integer vertices previously defined in the file. The order of the vertices is not arbitrary, they must be listed in a continuously clockwise or counter clockwise order, not a "bow tie" order. Not all four vertex polygons form a closed rectangle; for more complex shapes use multiple triangle commands instead. The "isreflective" flag is -1 if the surface absorbs, 1 if the surface reflects, or 0 if the surface refracts. Using this flag allows some rectangles to be reflective, and others to be refractive or absorptive, within the same Polygon

Object or face. However, the isreflective flag is not considered in the object produced by Boolean operations with the Polygon Object.

The value for face defines which face the rectangle belongs to. If the face number is omitted; face 0 is assumed.

Example:

R 1 2 3 4 1 0

This would define a reflective rectangle connecting vertices 1, 2, 3, and 4, belonging to face 0.

Maximum triangles in the polygon object

There is no fixed upper limit on the number of triangles a polygon object may contain. The ultimate limit is determined by the amount of real or virtual RAM the computer has available. Each triangle takes up about 100 bytes of memory. However, Zemax often maintains multiple copies of the lens data simultaneously, and so a good rule of thumb is that Zemax will need 500 bytes of RAM for each triangle. A 2000 triangle object will require about

1 Mb of free RAM. A more practical limit is computer speed; Zemax will slow down noticeably if the number of triangles becomes very large.

Example POB file

Here is the complete text of a POB file defining a cube of refractive material. The file is included as the example

CUBE.POB. All 8 sides of the cube belong to face 0.

! A cube

! front face vertices

V 1 -1 -1 0

V 2 1 -1 0

V 3 1 1 0

V 4 -1 1 0

! back face vertices

V 5 -1 -1 2

V 6 1 -1 2

V 7 1 1 2

V 8 -1 1 2

! Front

R 1 2 3 4 0 0

! Back

R 5 6 7 8 0 0

! Top

R 4 3 7 8 0 0

! Bottom

R 1 2 6 5 0 0

! Left side

R 1 4 8 5 0 0

! Right side

R 2 3 7 6 0 0

The 8 "V" commands define the vertex coordinates of the 8 corners of the cube. The 6 "R" commands define each of the 6 faces of the cube. Note the width of the faces is 2 units, and the z coordinates of the back face vertices is defined to be 2 units; so the shape is a perfect cube. All the coordinates are relative to the object reference point; which in this case will be the center of the front face. To place the reference point at the center of the cube, change the vertex definitions to:

V 1 -1 -1 -1

V 2 1 -1 -1

V 3 1 1 -1

V 4 -1 1 -1

V 5 -1 -1 1

V 6 1 -1 1

V 7 1 1 1

V 8 -1 1 1

For more examples see the POB files provided with Zemax in the <objects>\Polygon Objects folder (see "Folders").

Ray Rotator

The Ray Rotator is an imaginary device that rotates all incident rays. The object is a plane surface with either an elliptical or a rectangular shape. At the point a ray intersects the rotator, a "ray" coordinate system is created. The X, Y, and Z directions of this ray coordinate system initially are parallel to the local X, Y, and Z directions of the rotator, but the ray coordinate system is centered on the ray intercept point rather than the surface vertex.

The incident ray direction cosines are then rotated around the Z axis of the ray coordinate system. The rotation direction uses the right hand rule. This Z rotation also rotates the ray coordinate Y axis by the same angle. The rays are then rotated around the new ray coordinate Y axis resulting from the first Z direction rotation. If the rays are polarized, the polarization vector is also rotated. Note the ray's position is never modified; only the ray direction cosines.

The rotator object will not refract, absorb, attenuate, alter optical path or phase, or otherwise modify any property of the ray other than its direction and the direction of any associated electric field data. If the material is set to MIRROR, this object will reflect rays, and the rotation is applied after the reflection. Because the phase and optical path length are not modified, wavefront phase is not preserved by this object and thus only incoherent computations may be meaningful on the beam subsequent to the ray rotation.

The defining parameters are:

1/2: The X/Y Half Widths. Make positive for a rectangular shape, negative for an elliptical shape.

3/4: The Z/Y direction rotation angles in degrees.

Rectangular Corner
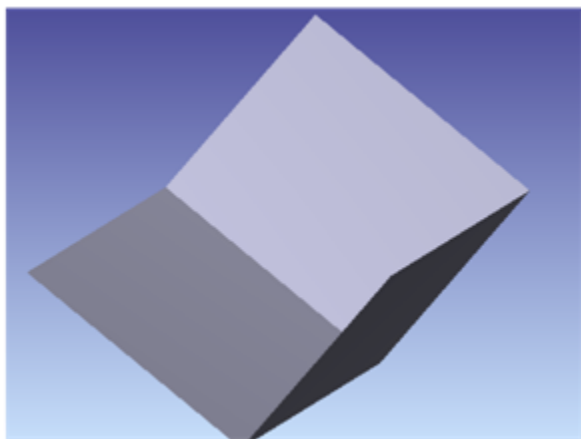
Rectangular corners are defined by 1 parameter:

1: The full width of the squares.

Rectangular corners are composed of 3 squares which meet at 90 degrees. The squares are aligned in the positive XY, XZ, and YZ planes. Each square is of dimension X by X.

The reference point is the point where all 3 squares touch. Face Numbers: All faces Face 0.

Rectangle
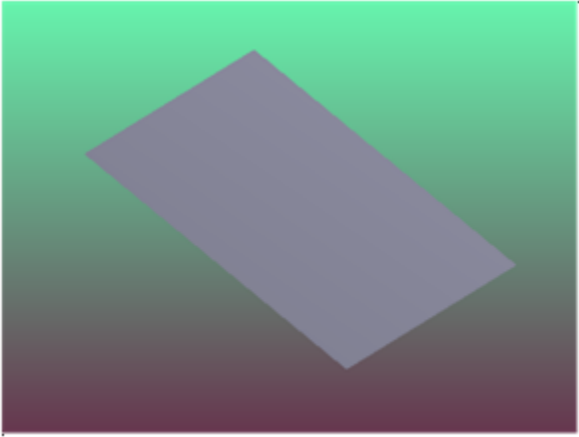
A rectangle is a flat surface defined by 2 parameters:

1: The X half width.

2: The Y half width.

Rectangles are flat and reside entirely within the local XY plane and are placed at the local $Z = 0$ coordinate.

The reference point is the center of the rectangle. Face Numbers: All faces Face 0.

Rectangular Pipe

Rectangular pipes are defined by 9 parameters:

1: The X half width of the open front face.

2: The Y half width of the open front face.

3: The Z length of the pipe along the local Z axis.

4: The X half width of the open rear face.

5: The Y half width of the open rear face.

6-7: The X, Y angle tilt in degrees of the front face.

8-9: The X, Y angle tilt in degrees of the rear face. Rectangular pipes are 4 sided boxes. The front and rear faces are open. This object is typically used as a rectangular light pipe.

The reference point is the center of the front open face. Face Numbers: All faces Face 0.

See "Comment about tilted faces" , those comments apply to this object as well.

Rectangular Pipe Grating

This object is the same shape as the Rectangular Pipe and uses the same first 9 parameters. However, the grating version adds a linear diffraction grating on all four sides. There are two additional parameters:

10: The grating line frequency in lines/micrometer on the side faces.

11: The diffraction order. Multiple orders may be specified, see "Diffraction tab" .

The grating consists of equally spaced lines perpendicular to the local z axis, lying on each of the four faces. The grating frequency is the lines per micrometer along the z direction; projected down on to either the XZ or YZ plane. Note the grating exists on all four sides of the pipe. The reference point is the center of the front open face.

For important information on diffractive objects, see "Diffraction from NSC objects" . Face Numbers: All faces Face 0.

Rectangular Roof
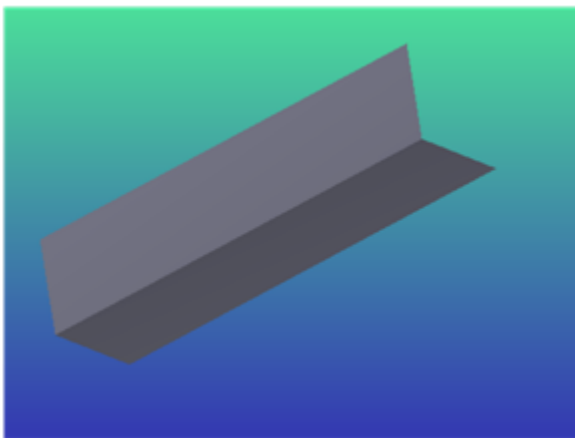
There are 3 parameters used to define a rectangular roof:

1: The X half width.

2: The Y half width.

3: The angle between the two rectangles.

A rectangular roof is composed of two rectangles which meet at an angle.

The reference point is the midpoint of the line joining the two rectangles. Face Numbers: All faces Face 0.

Rectangular Torus Surface
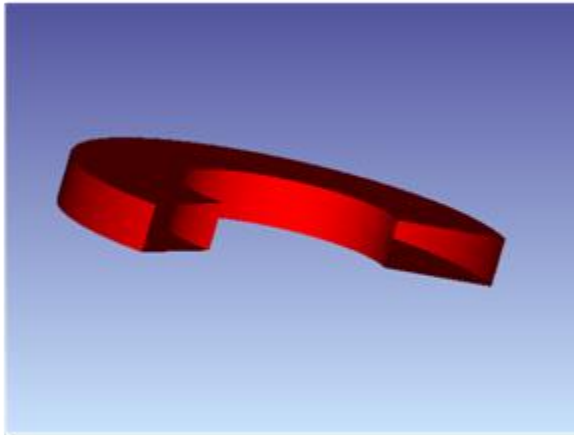
A rectangular torus is a surface formed by rotating a rectangle about a displaced axis. The rotation about the displaced axis may be over a full 360 degrees; or just some subset of that angular range. See also the discussion of the Rectangular Torus Volumefor modeling refractive solid torus shapes.

The rectangular torus surface is defined by 6 parameters:

1: The outer radius of the torus, Rout.

2: The inner radius of the torus, Rin.

3: The start angle of the torus, θ_1.

4: The stop angle of the torus, θ_2.

5: The thickness of the torus, Ty.

The rectangle lies in the YZ plane with the center at $x = 0$, $y = 0$, $z = (Rout+Rin)/2$. This position of the rectangle corresponds to the rotation angle $\theta = 0$. The angles of rotation are about the Y axis and must meet this condition:

$$0 \le \theta_1 \le \theta_2 \le 360$$

There is also the restriction that Rout > Rin > 0 and Ty > 0. The reference coordinate is the center of the axis of rotation. Face Numbers: All faces Face 0.

Rectangular Torus Volume

This object is essentially identical to the Rectangular Torus Surface, except the ends of the torus are closed to make a solid volume. This allows the object to made of a refractive material. See the description of the Rectangular Torus Surface for details. Face Numbers: Start angle end cap Face 1, stop angle end cap Face 2, all other faces Face 0.

Rectangular Volume
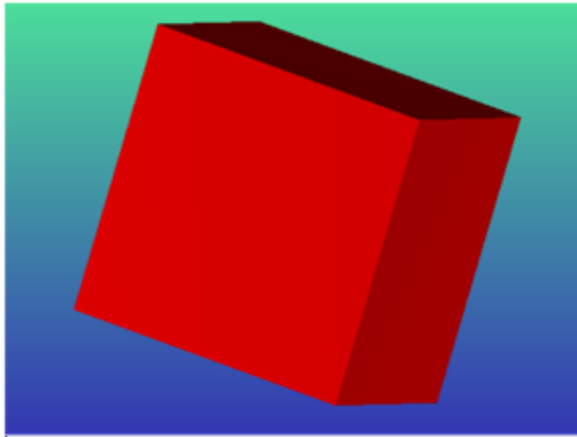
There are 9 parameters used to define the rectangular volume:

1: The X half width of the front face.

2: The Y half width of the front face.

3: The Z length of the volume along the local Z axis.

4: The X half width of the rear face.

5: The Y half width of the rear face.

6: The front face tilt angle in degrees along X.

7: The front face tilt angle in degrees along Y.

8: The rear face tilt angle in degrees along X.

9: The rear face tilt angle in degrees along Y.

Rectangular volumes are usually 6 sided solids. However, shapes such as pyramids and wedges may be created by setting one or more of the parameters to zero.

When performing Boolean operations with rectangular volumes, or exporting rectangular volumes in CAD format, any face with a width of less than the export tolerance will be removed.

The reference point is the center of the front face. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Comment about tilted faces

Note that if the front and rear face X half widths are not the same, and there is a tilt along the Y direction on either the front or rear face, then the four corners on the sides of the "box" are no longer coplanar. This yields an odd shape that Zemax models as two triangles, with a crease along the diagonal. A similar condition occurs if the front and rear Y half widths are different and a tilt along X is defined on either face. Although no error message is issued in these cases, the resulting object model should be carefully studied to confirm the desired shape is generated.

Rectangular Volume Grating

This object is the same shape as the Rectangular Volume and uses the same first 9 parameters. However, the grating version adds a linear diffraction grating on four faces of the object: the top, bottom, left, and right faces. There is no grating on the front or back faces. There are two additional parameters:

10: The grating line frequency in lines/micrometer on the side faces.

11: The diffraction order. Multiple orders may be specified, see "Diffraction tab" .

The grating consists of equally spaced lines perpendicular to the local z axis, lying on each of the four faces. The grating frequency is the lines per micrometer along the z direction; projected down on to either the XZ or YZ plane. Note the grating exists on all four sides of the volume. The reference point is the center of the front face.

For important information on diffractive objects, see "Diffraction from NSC objects" . Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

ReverseRadiance Detector
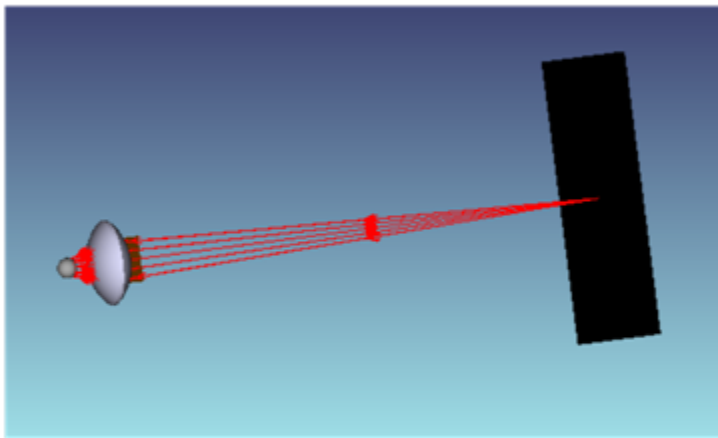
This feature is only available in the Premium edition of Zemax.

The ReverseRadiance Detector is similar to the Detector Rectangle object (see "Detector Rectangle object"), but uses a reverse ray tracing technology to determine the flux hitting the detector rather than passively summing the flux from rays traced in the conventional manner.

For the ReverseRadiance Detector to work properly, it must have an associated ReverseRadiance Target object, as well as objects defined as source boundaries and Source File objects with ray sets generated from Radiant Source Model files.

For a complete description and important information on the proper use of this feature see "ReverseRadiance ™ Analysis" .

ReverseRadiance Target

 This feature is only available in the Premium edition of Zemax.

This object is used in conjunction with the ReverseRadiance Detector to compute flux on a detector using a reverse ray tracing technique. For a complete description and important information on the proper use of this feature see "ReverseRadiance™ Analysis" .

Slide

Slides are color RGB transparencies defined by either a BMP, JPG, or PNG format graphics file.

Slides use the following parameters:

1: The width in lens units of the image.

2: The pixel aspect ratio, defined as height/width. Most bitmaps use a value of 1.0 for the pixel aspect ratio. The height of the slide will be determined by number of pixels, the slide width, and the pixel aspect ratio.

This object can be used to model radiant scenes by placing a source behind the slide. For example, to make a Lambertian radiant scene, place a collimated rectangular source (see "Source Rectangle") behind the slide, then set the slide to have a Lambertian scattering property (see "Scattering").

The transmission of the slide uses the same algorithm as described in "Slide" . Face Numbers: All faces Face 0.

Sphere
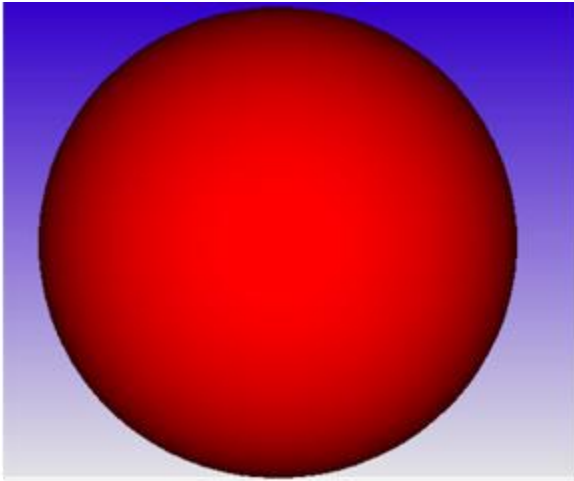
Spheres are defined by two parameters:

1: The radius of the sphere.

2: A flag to indicate if the sphere is a solid volume or a hollow shell. If the "Is Volume?" parameter is zero, then Zemax assumes the sphere defines a hollow shell. Rays must either reflect or absorb from such a surface; refraction is not allowed. If the "Is Volume?" parameter is any non-zero value, then Zemax assumes the sphere is a closed volume. The volume may be reflecting, refracting, or absorbing.

This object can be used to model bubbles by placing a sphere within a glass volume and setting the material type to blank (air) or the name of a defined glass which describes a gas.

The reference point is the center of the sphere.

Face Numbers:All faces Face 0.

Standard Lens
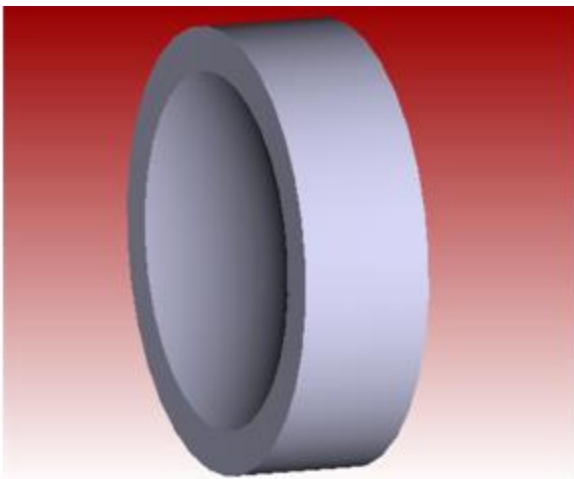
The standard lens is a lens composed of standard Zemax surfaces. Standard surfaces may be planes, spheres, conic aspheres, or hyperhemispheres. The standard lens is composed of 5 separate sections:

1) A standard shape front face.

2) A standard shape rear face.

3) An annular ring between the clear aperture of the front face and the edge of the front face.

4) An annular ring between the clear aperture of the rear face and the edge of the rear face.

5) A possibly tapered cylindrical surface connecting the edges of the front and rear faces of the lens.

9 parameters are used to define a standard lens:

1: The radius of curvature of the front face. Use zero for infinity (flat).

2: The conic constant of the front face.

3: The semi-diameter to the clear aperture of the front face. Use a negative value to yield the hyperhemispheric sag point.

4: The radial aperture to the edge of the front side of the lens.

5: The center-to-center thickness of the lens.

6: The radius of curvature of the rear face. Use zero for infinity (flat).

7: The conic constant of the rear face.

8: The semi-diameter to the clear aperture of the rear face. Use a negative value to yield the hyperhemispheric sag point.

9: The radial aperture to the edge of the rear side of the lens.

All 5 surfaces may refract, reflect, or absorb light, depending upon the material properties.

The reference point is the center of the front face of the lens. Face Numbers: Front face Face 1, back face

Face 2, all other faces Face 0.

Standard Surface
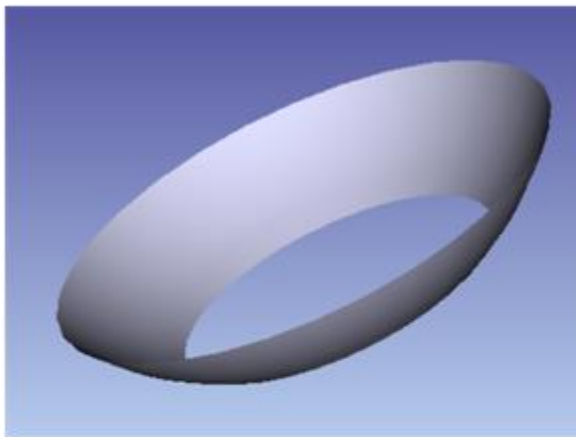
The standard surface object is very much like a standard surface in sequential Zemax. Standard surfaces include planes, spheres, conic aspheres, and hyperhemispheric spheres and aspheres.

The standard surface requires 4 parameters:

1: The radius of curvature. Use zero for infinity (flat).

2: The conic constant.

3: The semi-diameter to the maximum clear aperture. Use a negative value to yield the hyperhemispheric sag point.

4: The semi-diameter to the minimum clear aperture. This value must be positive. If greater than zero, this value creates a "hole" in the surface.

The surface is rotationally symmetric about the local z axis.

If the radius is negative, then the surface is concave towards the negative z axis. If the radius is positive, the surface is concave towards the positive z axis.

If the semi-diameter is negative, then the surface will become hyper-hemispheric, with the radial aperture of the open end equal to the absolute value of the semi-diameter.

Surfaces may reflect or absorb rays.

The reference point is the center of the surface. Face Numbers: All faces Face 0. This object supports user defined apertures, see "User defined apertures" . For a similar surface shape with more complex aperture shapes supported, see "Aspheric Surface 2" .

Swept Object
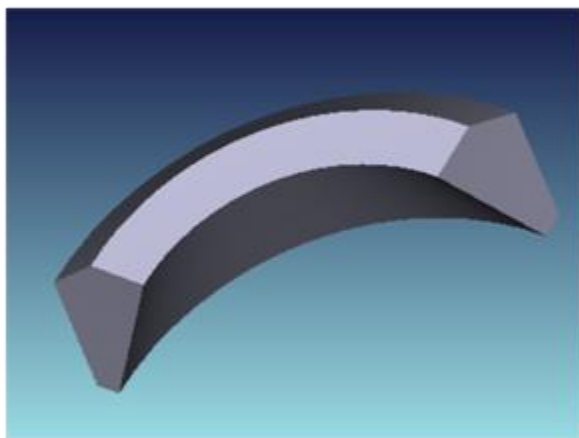
A swept object is created by sweeping a cross section of a parent object around an arbitrary axis to form a new shape. The parent object may be any surface or solid volume. A cross section of the parent object is formed by the intersection of the parent and a "slicing" plane. The slicing plane is defined by the axis of rotation, and a vector that points away from this axis of rotation. The resulting cross section may be swept in angle from -360 to +360 degrees, with the restriction that the absolute value of the angle must be at least 0.1 degrees. Swept objects are defined by the following parameters:

1: The parent object number. This number must precede the

swept object number in the editor.

2,3,4: The center x, y, and z coordinates in the unrotated, unshifted coordinate system of the parent object. These data define a point that lies on the axis of rotation.

5,6,7: The axis x, y, and z direction cosines in the coordinate system of the parent object. These cosines are normalized to determine the vector pointing along the axis of rotation. The axis of rotation must not intersect the object, or an invalid solid will be generated. Generally this condition cannot be automatically detected and no warning will be issued.

8,9,10: The plane x, y, and z direction cosines in the coordinate system of the parent object. These cosines are normalized to determine a vector pointing away from the axis of rotation. The slicing plane is defined as the plane that contains both the axis of rotation and the plane vector. Note the plane vector is not orthogonal to the slicing plane, but lies within the plane. The plane vector must not be parallel to the axis vector.

11: The sweep angle in degrees. The sweep begins at the plane.

12: The end caps option. Use 0 for no end caps, 1 for first end cap only, 2 for second end cap only, and 3 for both end caps. If the sweep angle is 360 degrees no end caps are formed, regardless of this setting. If the sweep angle is less than 360 degrees, and the "Is Volume" flag is set to a non-zero value, then the end caps setting is ignored, and both end caps are automatically added.

13 - 15: Unused.

16: Is Volume?: If the resulting shape is an enclosed volume, set this parameter to 1, otherwise, the object is a shell and this parameter should be set to zero.

Tabulated Faceted Radial
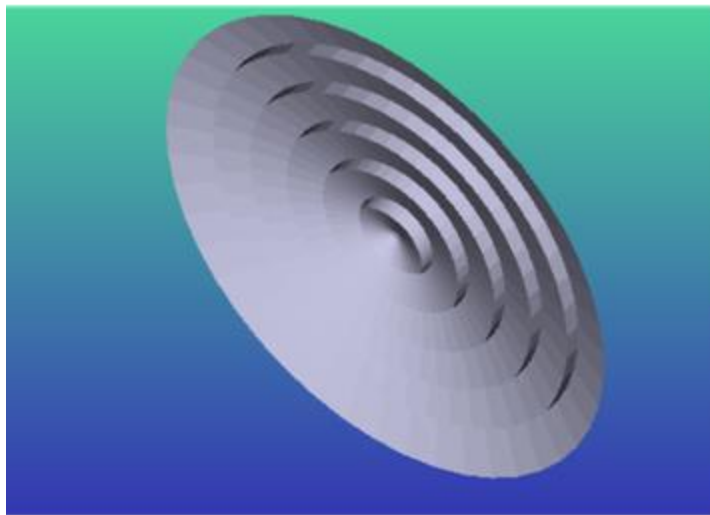
A tabulated object is based upon coordinates defined in a file. The file must reside in the <objects>\Tabulated Objects folder (see "Folders") and end in the extension TOB. The coordinates represent the starting and ending points of facets. A figure of revolution is generated by replicating a specified number of facets over some angular range. The axis of revolution is the local Z axis.

The TOB file format is two columns of data separated by either one or more spaces or one or more tab char- acters. A sample TOB file might look like this:

1.5 3.5

2.2 4.5

3.0 5.5

3.0 6.0

The first number of each pair is the local Y coordinate. This value must be zero or positive. The second value is the local Z coordinate. Each pair of values after the first pair represent a "zone". If there are 6 pairs of numbers, then there are 5 zones to the object. There is a maximum of 246 zones per object. Multiple objects may be used if more zones are required.

Zemax will generate facets which approximate a smooth surface for each zone. The facets can cover any fraction of a full circle, defined by a start and stop angle. To make a full figure of revolution, the start angle should be set to 0.0 degrees, and the stop angle to 360.0 degrees. Both angles must be zero or positive and be less than or equal to 360.0 degrees.

The number of facets generated over this angular range can be specified independently for each zone; so there may be 40 facets in the first zone, 80 in the second, 50 in the third, etc. If the number of facets changes from zone to zone, small gaps in the part may form and these gaps will generate geometry errors. If the gaps are small, the rays which trigger these errors may be ignored; see "The Ray Trace Control" for more information.

The parameters used to define the object are:

1: A scale factor. All vertices in the TOB file are multiplied by this parameter.

2: A flag to indicate if the TOB file defines a volume or a surface. If the "Is Volume?" parameter is zero, then Zemax assumes the TOB file defines an open surface. Rays must either reflect or absorb from such a surface; refraction is not allowed. If the "Is Volume?" parameter is any non-zero value, then Zemax assumes the TOB file defines a closed volume. The volume may be reflecting, refracting, or absorbing.

3: Start angle. The angle in degrees to begin the revolution of the TOB defined coordinates.

4: Stop angle. The angle in degrees to end the revolution of the TOB defined coordinates.

5: Zone 1 facets. The number of facets between the start and stop angles for the first zone.

4+n: Zone n facets. The number of facets between the start and stop angles for the nth zone.

If the "Is Volume" flag is set, the TOB file must define an object that will be a closed volume upon rotation. This requires that the object be rotated fully around 360.0 degrees. Fully closed volumes defined by TOB files may be used to model faceted approximations to true Fresnel lenses.

Face Numbers: All faces Face 0.

Tabulated Faceted Toroid
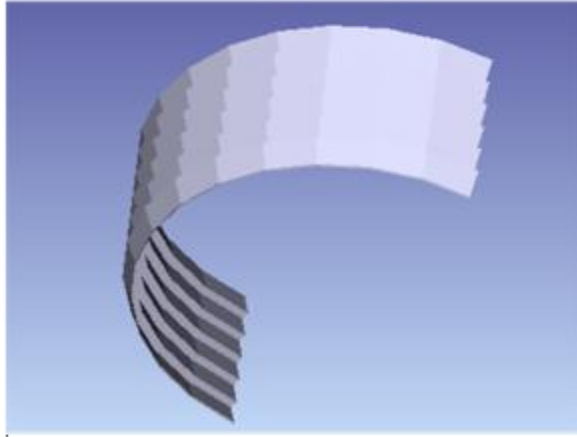
A tabulated object is based upon coordinates defined in a file. The file must reside in the <objects>\Tabulated Objects folder (see "Folders") and end in the extension TOB. The coordinates represent the starting and ending points of facets. A figure of rotation is generated by replicating a specified number of facets over some angular range. The axis of rotation is an axis parallel to the local Y axis offset by a specified radius. If the radius is set to zero, then a surface with cylindrical symmetry is generated instead of a torus.

The TOB file format is two columns of data separated by either one or more spaces or one or more tab characters. A sample TOB file might look like this:

1.5 3.5

2.2 4.5

3.0 5.5

3.0 6.0

The first number of each pair is the local Y coordinate. This value may be negative, zero, or positive. The second value is the local Z coordinate. Each pair of values after the first pair represent a "zone". If there are 6 pairs of numbers, then there are 5 zones to the object. There is a maximum of 246 zones per object. Multiple objects may be used if more zones are required.

Zemax will generate facets which approximate a smooth surface for each zone. The facets can cover any fraction of a full circle, defined by a start and stop angle. To make a full figure of revolution, the start angle should be set to -180.0 degrees, and the stop angle to 180.0 degrees. Both angles must have an absolute value less than or equal to 180.0 degrees.

The number of facets generated over this angular range can be specified independently for each zone; so there may be 40 facets in the first zone, 80 in the second, 50 in the third, etc.

The parameters used to define the object are:

1: A scale factor. All vertices in the TOB file are multiplied by this parameter.

2: Radius of rotation. If positive, then the axis of rotation is in the positive local Z direction, parallel to the local Y axis, in the YZ plane. If negative, then the axis of rotation is in the negative local Z direction, parallel to the local Y axis, in the YZ plane. If zero, then a cylinder results. In the special case of zero radius, the number of facets parameters are

ignored (since a single facet perfectly models a flat plane) and the start and stop angle are interpreted as start and stop x coordinate in lens units.

3: Start angle. The angle in degrees to begin the rotation of the TOB defined coordinates, unless the radius of rotation is zero; in this case the start angle defines the starting x coordinate in lens units.

4: Stop angle. The angle in degrees to end the rotation of the TOB defined coordinates, unless the radius of rotation is zero; in this case the stop angle defines the stopping x coordinate in lens units.

5: Zone 1 facets. The number of facets between the start and stop angles for the first zone.

4+n: Zone n facets. The number of facets between the start and stop angles for the nth zone. Face Numbers: All faces Face 0.

Tabulated Fresnel Radial

This object is nearly identical to the tabulated faceted radial object. The key difference is that the radially symmetric surfaces are smooth, rather than faceted. The maximum number of points is also much larger at

50,000. See "Tabulated Faceted Radial" for a description of this type of object and the tabulated object file format. Face Numbers: All faces Face 0.

Toroidal Hologram
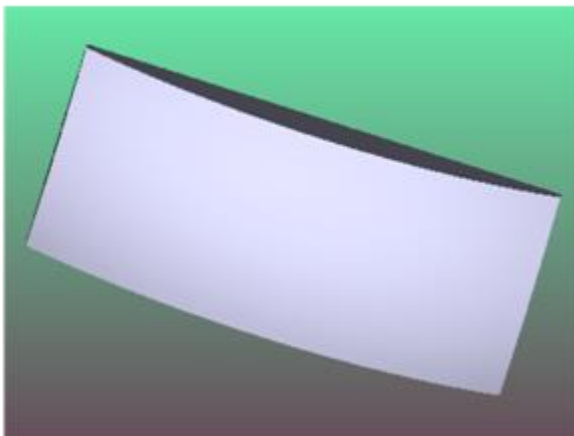
A Toroidal Hologram consists of a rectangular, circular, or elliptical lens with possibly aspheric toroidal surfaces on the front and back faces and an optically fabricated hologram on the front face. A toroidal surface is defined

by a curve in the YZ plane which is then rotated about an axis parallel to the Y axis but displaced by a distance

R; the radius of rotation. The curve in the YZ plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12}$$

where c is the reciprocal of the radius of curvature in YZ plane.

The hologram on the front face is defined in a manner similar to the Hologram 1 and Hologram 2 sequential surface models (see "Hologram 1" for a description of these surfaces). The Toroidal Hologram is defined by 32 parameters:

1: The Radial Height of the lens in the y direction.

2: The X Half-Width. If this parameter is zero, then the outer boundary of the lens will be a circle with a radial size equal to the Radial Height. If this parameter is positive, the outer boundary of the lens will be rectangular. If this parameter is negative, the outer boundary of the lens will be elliptical.

3: The Thickness of the lens along the local Z axis.

4-5: Unused.

6, 7, 8: The radius of rotation, radius of curvature, and conic for the front face.

9-14: The coefficients on the powers of y for the front face.

15, 16, 17: The radius of rotation, radius of curvature, and conic for the back face.

18-23: The coefficients on the powers of y for the back face.

24: Holo Type: Use 1 for Hologram type 1 (both sources converging/diverging) or 2 for Hologram type 2 (one source converging, one diverging) See "Hologram 1" .

25: Order: The diffraction order to use. Multiple orders may be specified, see "Diffraction tab" .

26: Construction Wavelength: The wavelength in micrometers used to fabricate the hologram.

27-32: The X, Y, and Z coordinates in lens units of the construction points relative to the vertex of the front face of the Hologram.

To make any of the 4 radii flat; use a value of zero. Note a cylinder lens results if the radius of rotation is set to zero. If the order is zero or if the ray totally internally reflects at the hologram boundary, no hologram diffraction is computed.

For important information on diffractive objects, see "Diffraction from NSC objects" .

The reference coordinate is the center of the front face. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

Toroidal Lens
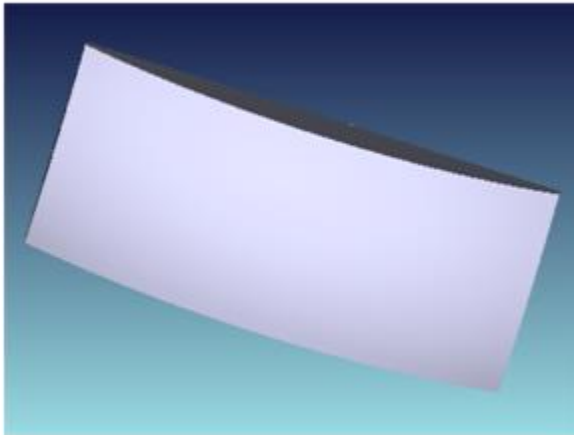
A toroidal lens consists of a rectangular, circular, or elliptical lens with possibly aspheric toroidal surfaces on the front and back faces. A toroidal surface is defined by a curve in the YZ plane which is then rotated about an axis parallel to the Y axis but displaced by a distance R; the radius of rotation. The curve in the YZ plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14}$$

where c is the reciprocal of the radius of curvature in YZ plane.

The toroidal lens is defined by 23 parameters:

1: The X Half-Width in lens units.

2: The Y Half-Width in lens units.

3: Thickness

4-5: Unused.

6, 7, 8: The radius of rotation, radius of curvature, and conic for the front surface.

9-14: The coefficients on the powers of y for the front surface.

15, 16, 17: The radius of rotation, radius of curvature, and conic for the back surface.

18-25: The coefficients on the powers of y for the back surface.

To make either the radius of rotation or radius of curvature flat; use a value of zero. Note a cylindrical surface results if the radius of rotation is set to zero.

The reference coordinate is the center of the front face. Face Numbers: Face Numbers: Front face Face 1, back face Face 2, Side faces Face 0.

Toroidal Surface
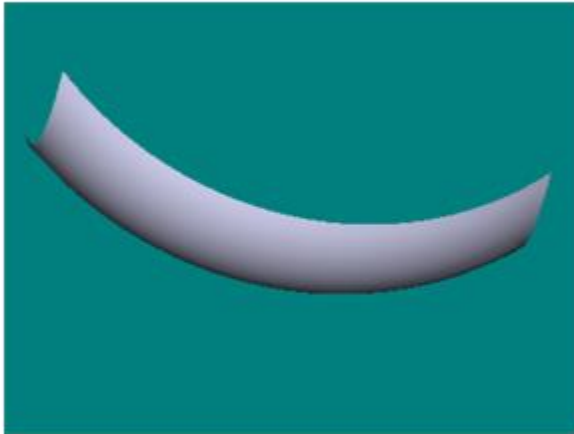
A toroidal surface consists of a rectangular surface with a possibly aspheric toroidal shape. A toroidal surface is defined by a curve in the YZ plane which is then rotated about an axis parallel to the Y axis but displaced by a distance R; the radius of rotation. The curve in the YZ plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2 y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14}$$

where c is the reciprocal of the radius of curvature in YZ plane.

The toroidal surface is defined by 14 parameters:

1: The X Half-Width in lens units.

2: The Y Half-Width in lens units.

3-4: Unused.

5, 6, 7: The radius of rotation, radius of curvature, and conic for the surface.

8-14: The coefficients on the powers of y for the surface.

To make either the radius of rotation or radius of curvature flat; use a value of zero. Note a cylindrical surface results if the radius of rotation is set to zero.

The reference coordinate is the center of the front face. Face Numbers: All faces Face 0.

Torus Surface
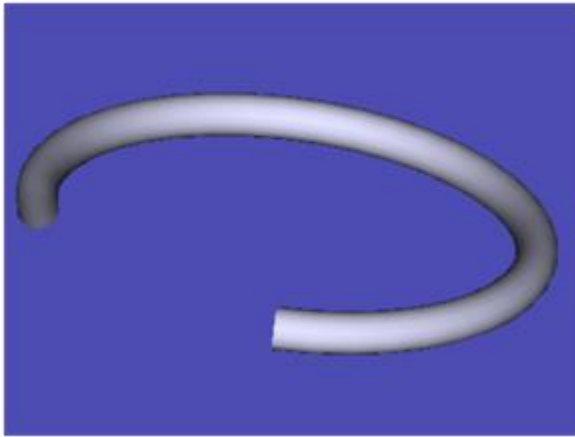
A torus is a circle rotated about a displaced axis. The rotation about the displaced axis may be over a full 360 degrees; or just some subset of that angular range. This object can be used to model optical fibers or curved light pipes. See also the discussion of the Torus Volume for modeling refractive solid torus shapes.

The torus surface is defined by 4 parameters:

1: The radius of rotation about the Y axis of the circle, R.

2: The radius of the circle, r.

3: The start angle of the torus, θ_1.

4: The stop angle of the torus, θ_2.

The circle lies in the YZ plane with the center of the circle at x = 0, y = 0, and z = R. This position of the circle corresponds to the rotation angle $\theta = 0$. The angles of rotation are about the Y axis and must meet this condition:

$$0 \leq \theta_1 \leq \theta_2 \leq 360$$

There is also the restriction that R > r; otherwise, a closed volume or a smooth surface will not result. The reference coordinate is the center of the axis of rotation. Face Numbers: All faces Face 0.

Torus Volume

This object is essentially identical to the Torus Surface, except the ends of the torus are closed to make a solid volume. This allows the object to made of a refractive material. See the description of the Torus Surface for details. Face Numbers: Start angle end cap Face 1, stop angle end cap Face 2, all other faces Face 0.

Triangular Corner
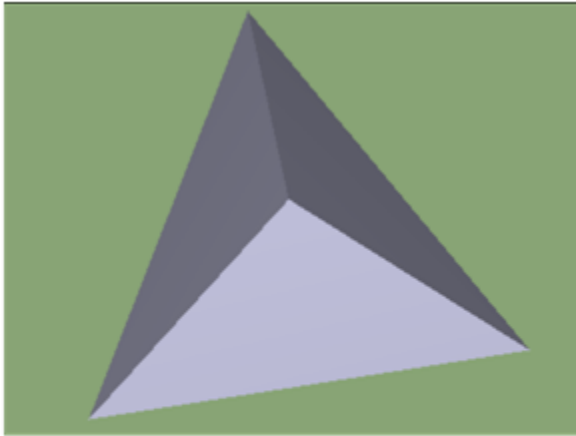
Triangular corners are defined by 1 parameter:

1: The full width of the short side of the triangle.

Triangular corners are defined by 3 triangles which meet at 90 degrees. The triangles are aligned in the positive XY, XZ, and YZ planes. Each surface is a 45-45-90 triangle, where the two short sides are of a length specified by parameter 1.

The reference point is the point where all 3 triangles touch. Face Numbers: All faces Face 0.

Triangle
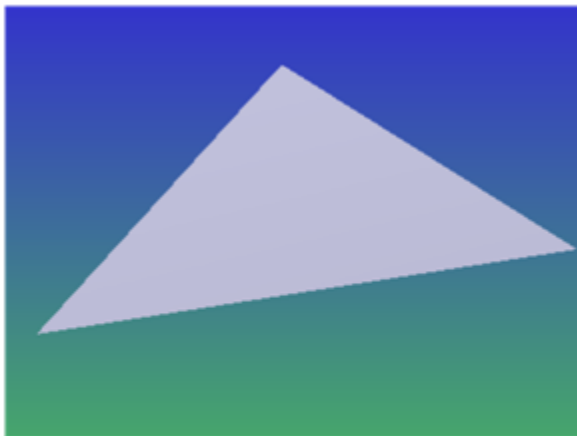
Triangles are defined by 3 points in the XY plane, which is a total of 6 parameters:

1: The X coordinate of vertex 1.

2: The Y coordinate of vertex 1.

3: The X coordinate of vertex 2.

4: The Y coordinate of vertex 2.

5: The X coordinate of vertex 3.

6: The Y coordinate of vertex 3.

The reference point is the (0, 0, 0) coordinate, which may not be part of the triangle at all; depending upon the values give for the vertex points. Face Numbers: All faces Face 0.

User Defined Object

There are several different ways to create user defined objects. For simple polygon based objects, such as prisms and other shapes with all flat faces, see "Polygon Object" . Objects may also be defined in an external CAD program and then imported into Zemax, see "CAD Part: STEP/IGES/SAT" and "CAD Part: STL" . There are also methods for user definitions for gradient index media ("Defining GRIN media for non-sequential ray tracing"), surface scattering properties ("Scattering"), bulk scattering properties ("Bulk scattering"), and diffraction ("Defining DLLs for ray splitting at diffractive surfaces").

The method of creating a user defined object described in this section involves the use of an external, user supplied program called a Dynamic Link Library or DLL. The advantages to defining an object using a DLL, rather than the other methods listed above, are:

- DLL defined objects generally ray trace much faster, and with much higher numerical precision, than objects imported from CAD programs.

- Any number of complex curved shapes may be combined in a single object, unlike the polygon object which only has flat faces.

- Objects may have a mixture of reflective and refractive curved faces, with user definable face names.

- The DLL description is inherently parametric, which means the object is dynamically regenerated when any defining property is modified. This allows interactive design, modification, and even optimization.

- User defined coating data, including detailed control over the complex amplitude reflection and transmis- sion coefficients is supported. Coating data may be ray position, cosine, or object coordinate dependent.

The entire definition for the object is contained within the DLL. The DLL computes the following data:

- The computation of basic data about the object, such as whether it is a solid volume or a hollow shell, and the names of all the used parameters.

- The determination of a list of triangles which approximate the shape of the object. These triangles are used to render the object as well as provide a "close first guess" for ray tracing purposes.

- The exact computation of the ray-surface intercept. Most of this code is actually inside Zemax; the DLL only needs to find the exact ray-surface intercept point given a very close guess. Zemax handles all the logic for non-sequential tracing, nesting, refraction, reflection, diffraction, optical path, scattering, etc.

- The user defined coating data used by the object, if any. The DLL or Zemax can compute coating data.

- The starting "safe" data to use for the various free parameters when the object is first created.

The DLL must include two functions:

UserObjectDefinition

UserParamNames

The UserObjectDefinition function is called to create a list of triangles which approximate the shape of the object. Zemax traces rays non-sequentially to the triangles to determine close intercepts, if any. Once a potential close intercept is found, the UserObjectDefinition function is called again to determine the exact intercept. This is implemented using an iterative routine provided in any of the sample object DLL source files. A single object may contain multiple complex curved faces, each with it's own defining function and iteration routines.

The best way to learn the use of Object DLLs is to study an existing DLL and modify it as needed. The sample DLLs provided with Zemax include extensive documentation and comments on the data formats and DLL functions; see any of the sample object code files for examples.

All Object DLLs must be placed in the <program>\DLL\Objects folder.

Face Numbers: All faces have user definable Face numbers.

If you require a User Defined Object DLL, but do not wish to write the DLL yourself, contact Zemax Technical Support for a quote on developing a custom DLL to meet your requirements. We have considerable experience in developing ray tracing algorithms, and can generally write user defined object DLL code for a small fee, often on very short notice.

Sample Object DLLs

The following sample object DLLs and source code are included with Zemax.

SAMPLE OBJECT DLLS

CoatingSample	A rectangular surface, with user definable dimensions in x, y. This sample shows how to implement user defined coating data. The transmission of this rectangle is 1.0 outside of a circle of radius "a", and a user defined value inside of the circle.
EllipticalVolume	An elliptical solid volume, with user definable dimensions in x, y, and z. This DLL demonstrates the creation of a list of triangles to approximate the object shape, naming object faces, and exact ray-surface iteration to arbitrary surface description functions.
HalfCylinder	A half cylinder, with user definable radius and length. This DLL demonstrates the creation of a list of triangles to approximate the object shape, naming object faces, and exact ray-surface iteration to arbitrary surface description functions.

Wolter Surface
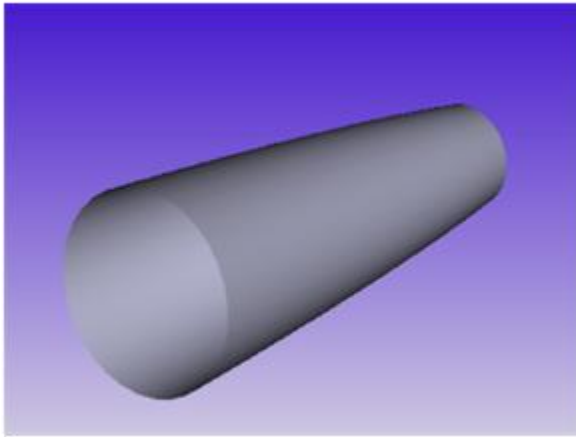
The Wolter Surface is a radial symmetric shell object defined by the following parameters:

1: The length along the Z axis.

2: The radial aperture at z = 0, R0.

3: The angle in degrees.

4-7: The polynomial coefficients on Z^2, Z^3, Z^4, and Z^5.

The object starts at Z = 0 and ends at the specified length.

A Wolter Surface is commonly used in X-Ray telescope optics. The surface is a hollow shell and may be nested within other shells. The shell has a radius that is a function of the z coordinate given by the following expression:

$$R^2(z) = R_0^2[1 + 2z\tan\theta + a_2z^2 + a_3z^3 + a_4z^4 + a_5z^5]$$

Face Numbers: All faces Face 0.

Zernike Surface
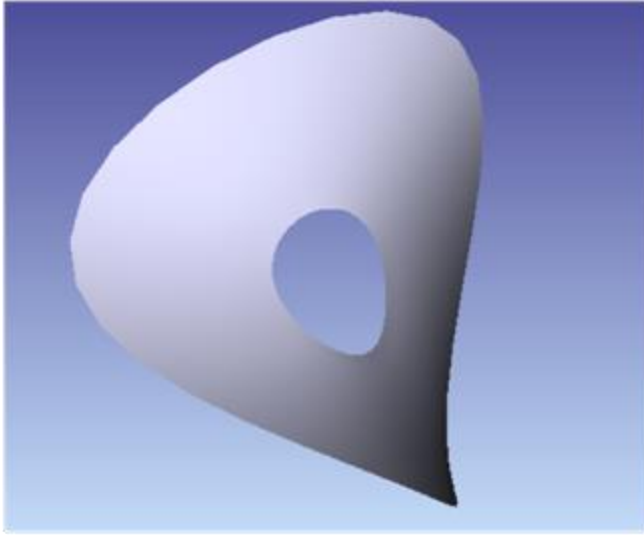
A Zernike surface is defined by the following sag equation:

$$z = \frac{cr^2}{1 + \sqrt{1-(1+k)c^2r^2}} + \sum_{i=1}^{8} \alpha_i r^{2i} + \sum_{i=1}^{N} A_i Z_i(\rho, \varphi)$$

where c is the curvature of the surface, k is the conic constant, the α terms are aspheric coefficients, N is the number of Zernike coefficients in the series, A_i is the coefficient on the *i* Zernike Standard polynomial, r is the radial coordinate in lens units, ρ is the (optionally decentered) normalized radial coordinate, and ϕ is the (optionally decentered) angular coordinate. The Zernike Standard polynomials are defined in the table given in "Zernike Standard Coefficients". The surface supports specification of both a minimum and a maximum radial aperture; so annular surfaces may be defined.

The following parameters are used to define the Zernike surface:

1: The radius of curvature. If this value is zero, then the curvature is assumed to be zero.

2: The conic constant k.

3: The maximum radial aperture in lens units. This aperture is also used to normalize the radial coordinates used in the Zernike polynomial expansion, unless the minimum radial aperture is negative, see below.

4: The minimum radial aperture in lens units. This value may be zero. If this value is negative, then the minimum radial aperture is set to zero, and the absolute value of the parameter is used as the normalization radius. This feature allows a normalization radius distinct from the maximum radial aperture.

5-6: The decenter of the Zernike terms with respect to the base asphere in lens units.

7-14: The even radial aspheric coefficients α .

15: The number of terms to use in the Zernike expansion.

16-247: The Zernike coefficients.

Face Numbers: All faces Face 0. This object supports user defined apertures, see "User defined apertures".

Non-Sequential Sources

Sources include points, ellipses, rectangles, volumes, data files, and user defined types. Any source may be placed inside of any object, or not in any object, but not both (a source may not straddle an object boundary).

Summary of NSC Sources

Source Diffractive	A source with the far-field diffraction pattern of a defined UDA.
Source Diode	An array of diodes with separate X/Y distributions
Source DLL	A source defined by an external user supplied program.
Source Ellipse	An elliptical surface that emits light from a virtual source point.
Source EULUMDAT File	A source defined by lamp data in an EULUMDAT format file.
Source Filament	A source in the shape of a helical filament.
Source File	A user defined source whose rays are listed in a file.
Source Gaussian	A source with a Gaussian distribution.
Source IESNA File	A source defined by lamp data in an IESNA format file.
Source Imported	A source defined by the shape of an imported object.
Source Object	A source defined by the shape of another object.
Source Point	A point source that radiates into a cone. The cone may be of zero width or be extended up to a full sphere if desired.
Source Radial	A radial symmetric source based upon a spline fit of arbitrary intensity vs. angle data.
Source Ray	A point source aligned with direction cosines.
Source Rectangle	A rectangular surface that emits light from a virtual source point.
Source Tube	A source in the shape of a cylindrical tube.
Source Two Angle	A rectangular or elliptical surface that emits light into a cone with distinct angles in the X and Y directions.
Source Volume Cylinder	A volume source in the shape of a cylinder with an elliptical cross section.
Source Volume Ellipse	A source in the shape of an elliptical volume.
Source Volume Rectangle	A volume source in the shape of a rectangle.

All source type objects have the same definition for parameters 1 through 5. These parameters are:

1: # Layout Rays: Defines how many random rays to launch from the source when creating layout plots.

2: # Analysis Rays: Defines how many random rays to launch from the source when performing analysis.

3: Power (units): Power is the total power over the defined range of the source. The power units are specified by the system source units. See "Source Units" for details.

4: Wavenumber: The wavenumber to use when tracing random rays. Zero means polychromatic; which chooses ray wavelengths randomly with the weighting defined on the wavelength data editor.

5: Color #: The pen color to use when drawing rays from this source. If zero, the default color will be chosen. The other parameters have source type specific meanings as described in the follow sections.

Placing sources inside objects

By default, Zemax assumes a source is placed in the global medium of the non-sequential group. However, sources may be placed entirely inside any solid object, or inside any number of nested solids. The object the source is placed within must precede the source in the NSC Editor. Two steps are required to properly place a source inside of a solid object:

1) The source location and size must be correctly set such that all rays initiate from inside the object. This means the source cannot extend outside of the object in which it is placed.

2) The object in which the source is placed must be defined using the "Inside Of" data column of the source. There are two ways to define the object in which the source is placed: either by absolute or relative reference.

If a positive number greater than zero is specified in the Inside Of column, then the source is inside the specified object. This is an "absolute" reference. If the "Inside Of" object number is negative, then the source is inside of the object whose number is determined by adding the current object number to the negative "Inside Of" object number. This is a "relative" reference object. For example, if the "Inside Of" value is -3 on source object 8, the source must be placed inside of object 5 because 8 - 3 = 5. Relative references are particularly useful when copying and pasting groups of sources and objects; this is easiest if all the sources and objects in the group use relative references to the first object in the group. If the source is placed inside of an object which is inside of a second object; then the first object must define the second object number in the "Inside Of" data column for the first object. If there are more levels of nested objects, each object must define the object it has been placed inside of on the "Inside Of" data column. All of these nested definitions may be absolute, relative, or a mixture of both.

The source must be listed after all objects that the source is placed inside of in the NSCE.

Adding new source types

If a source is required that is not listed, please contact technical support to suggest the new type be added to Zemax. Some simple sources, such as Lambertian planes, may be simulated by irradiating a Lambertian scattering surface with one of the other source types. Many objects may also act as detectors, for example any polygon object may be a detector. See the discussion for information on the Source DLL and Source File for user defined sources.

Source Diffractive

The source diffractive calculates the wavefront in an aperture using Fraunhofer diffraction theory, this wavefront is then used to provide a set of direction cosines for the rays it generates. The spatial distribution of ray starting coordinates is randomly assigned within the aperture. All rays are assigned the same intensity value. The sum of these values is normalized to the power on the source, which is defined by parameter 3. The aperture is defined by a UDA file which is selected in the Type tab of the Object Properties window; any scaling of the aperture is also defined here.

The parameters used to define this source are:

1-5: See "Parameters common to all source objects" .

6: Sampling: Integer values that define the sampling. An input of 1 corresponds to 32x32 points in the grid up to a maximum input of 9, which corresponds to 8192x8912 points.

Fraunhofer Diffraction

For a short description on Fraunhofer diffraction theory see "Fraunhofer diffraction" . A more detailed discussion of this source may be found in the article entitled "The Source Diffractive Object" in the Zemax Knowledge Base (http://kb-en.radiantzemax.com/Knowledgebase/). For important information about sampling

issues see "Discussion of the FFT method and sampling issues" .

Use with LightningTrace

This source type is not supported by LightningTrace.

Source Diode

The source diode model can be used to define one diode, a 1D array of diodes, or a 2D array of diodes. Each diode has an intensity distribution given by:

$$I(\theta_x, \theta_y) = I_o e^{-2\left(\left(\frac{\theta_x}{\alpha_x}\right)^{2G_x} + \left(\frac{\theta_y}{\alpha_y}\right)^{2G_y}\right)}$$

where α_x is the XZ divergence angle in degrees, G_x is the "supergaussian" factor for the X direction, with similar definitions for the Y subscripted values. Note if G_x is 1.0, then a typical Gaussian distribution results. Both G_x and G_y must be greater than or equal to 0.01 and less than or equal to 50.0.

Most laser diode manufacturers specify the far field divergence angles as the full width of the distribution between the half power points, θ_{fwhm} . For a true Gaussian distribution ($G_x = 1$), setting the left hand side of the equation to 1/2 I_0, setting θ_y to zero, substituting for θ_x the value of ½ θ_{fwhm} , then solving for α_x gives

$$\alpha_x = \frac{\theta_{fwhm}}{\sqrt{2\ln(2)}}$$

Or

$$\alpha_x = (0.8493218)\theta_{fwhm}$$

For example, a diode with a θ_{fwhm} in the x direction of 10 degrees, the value for α_x would be 8.493218 degrees. A similar conversion applies in the y direction.

An astigmatism term may also be defined. This value must be positive, and represents the distance along the local Z axis from which the XZ distribution is measured. At the local XY plane at Z = 0, the resulting ray pattern is a line oriented along the local X axis.

If more than 1 diode is desired, the number x, number y, delta x, and delta y spacings may be defined. The diodes will be symmetrically placed in the x and y directions about the local coordinate origin.

If the astigmatism term is zero, the rays may emanate from a point, or from a line, or from a rectangular region. The spatial distribution of the ray source points is given by:

$$I(x,y) = I_o e^{-2\left(\left(\frac{x}{s_x}\right)^{2H_x} + \left(\frac{y}{s_y}\right)^{2H_y}\right)}$$

and

$$-W_x < x < W_x$$

where s_x is measured in lens units, Hx is the "supergaussian" factor for the X direction, with similar definitions for the Y subscripted values. Note if Hx is 1.0, then a typical Gaussian distribution results. Both Hx and Hy must be greater than or equal to 0.01. The spatial distribution terms are all ignored if the astigmatism term is not zero.

The parameters used to define this source are:

1-5: See "Parameters common to all source objects" .

6: Astigmatism: The distance to offset the XZ distribution in lens units.

7: X divergence in degrees (α_x).

8: X supergaussian factor (G_x).

9: Y divergence in degrees (α_y).

10: Y supergaussian factor (G_y).

11-12: Number of diodes in X/Y.

13-14: Spacing of diodes in X/Y in lens units.

15: The X half width of the rectangular region from which the rays emanate in lens units (W_x).

16: The Gaussian width of the spatial distribution in the X direction (s_x).

17: The X direction spatial distribution super Gaussian factor (H_x).

18: The Y half width of the rectangular region from which the rays emanate in lens units (W_y).

19: The Gaussian width of the spatial distribution in the Y direction (s_y).

20: The Y direction spatial distribution super Gaussian factor (H_y).

The angular distribution of the source is calculated using the X and Y divergence angles for the source (parameters 7, 9) and the X and Y super Gaussian factors (parameters 8, 20). The size and spatial distribution of the source is neglected. As such, the following input parameters have no impact on the LightningTrace results: astigmatism (parameter 6); X and Y half widths for the source (parameters 15, 18); X and Y Gaussian widths for the spatial distribution (parameters 16, 19); X and Y spatial distribution super Gaussian factors (parameters 17, 20). In addition, the array parameters for this source (parameters 11-14) are not accounted for by LightningTrace; regardless of the array parameter values, the source defaults to a single diode (unless otherwise defined under the Sources tab). If an array of diodes must be modeled, the array should be defined via the Array Type option under the Sources tab of the Object Properties dialog box (see "Modeling arrays of sources").

Source DLL

Although many types of built-in sources are included with Zemax, there are times when the best way to model a source is to define an algorithm to generate rays with the desired properties.

Zemax also supports user-defined sources as tables of rays; see the Source File discussion .

To define a source using a program, the algorithm which generates random rays must be written and compiled into a Windows Dynamic Link Library, or DLL. Numerous DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs" .

Source DLL parameters

Each DLL may use between zero and 30 user defined data values as parameters in the computation of the source properties. These values are defined by the DLL and are only used by the DLL.

Creating a new Source DLL

The DLL must include two functions: UserSourceDefinition UserParamNames

When launching a ray from a source modeled by a DLL, Zemax passes to the UserSourceDefinition function the source parameters, the wavelength, and other data. The UserSourceDefinition function then is required to compute the following values:

x, y, z: the starting coordinates for the ray

l, m, n: the starting direction cosines for the ray

i: the initial relative intensity of the ray (see "Comments on Source DLL relative intensity" below)

These values are returned to Zemax and are used to initiate the ray trace. The function UserParamNames is used to define the names of all used parameters. These names appear in the parameter columns of the source DLL object in the NSC Editor.

The best way to learn the use of Source DLLs is to study an existing DLL and modify it as needed. The sample DLLs provided with Zemax include extensive documentation and comments on the data format; see any of the sample source code files for examples.

All Source DLLs must be placed in the <program>\DLL\Sources folder.

Comments on Source DLL relative intensity

When Zemax traces a ray generated by a DLL, the initial power of the ray is given by the source total power, divided by the number of analysis rays, scaled by the relative weight. For this reason, the relative intensity of the ray as computed by the DLL must be set so that the average of all the relative intensities of all the analysis rays requested is 1.0. Note the number of analysis rays is passed as a parameter to the DLL. This condition is required because Zemax cannot normalize the entire collection of analysis rays to be traced without generating all the rays before tracing any.

The preferred way to normalize the rays is for the DLL to set the relative intensity of all rays to 1.0, and then choose rays according to the desired probability distribution of the source. This method implies that more rays will be selected with positions and direction cosines that correspond to more energy. The method for doing this is straightforward, and involves integrating the source over all spatial and direction space and using the normalized integral and a uniform random number generator to determine rays. The provided sample FIBER1 DLL illustrates the method.

An alternate way is to use non unity ray weights. This method requires that the DLL compute all the desired rays first, then internally normalize the rays to yield an average weight of 1.0. This method is conceptually simpler but is usually less efficient.

Use with LightningTrace

This source type is not supported by LightningTrace.

Sample Source DLLs

The following sample source DLLs and source code are included with Zemax

SAMPLE SOURCE DLL: FIBER1

A planar fiber model.

The rays emanate from a disk of radius R. The intensity profile is given by

$I(r) = A + Br^2 + C r^4$, where

$0.0 \, \delta \, r \, \delta \, R$.

Once the ray starting position is determined, the ray emanates from the surface into a cone whose numerical aperture depends upon the radial coordinate:

$NA(r) = D + Er^2 + Fr^4$.

Within the angular cone defined by the numerical aperture, the ray distribution is uniform. The parameters R, A, B, C, D, E, and F are all defined on the NSC Editor.

Source Ellipse

The Source Ellipse is a flat elliptical (and optionally annular) surface which emits rays. Although the origin of each ray launched lies on the surface of the ellipse with a uniform distribution, the distribution of the ray directions may be any one of these:

a) All rays appear to emit from a point placed anywhere relative to the source. The location of this point is defined in the parameter list below. When used in this mode, the source acts like an imaging point source.

b) Rays emit in a cosine distribution of the form

$$I(\theta) \approx I_o (\cos\theta)^{Cn}$$

where the exponent Cn may be any value between zero and 100, including non-integer values, and θ is the angle measured from the local surface normal. The larger Cn, the narrower the distribution becomes. If Cn = 0.0, the rays emanate in a hemisphere with equal probability in all directions. If Cn = 1.0, the distribution is Lambertian. Note this distribution is rotationally symmetric about the local Z axis.

c) Rays emit in a Gaussian distribution of the form

$$I(l, m) \approx I_0 e^{-(G_x l^2 + G_y m^2)}$$

where l and m are the direction cosines of the ray in the X and Y axis directions and Gx and Gy are constants. This form may be used to define a far field pattern that is different in the X or Y directions. The larger Gx and Gy are, the narrower the distribution becomes in the respective directions.

The nature of the ray distribution is defined by the parameter values. If Cn, Gx, and Gy, are all zero, then all rays will appear to emit from a virtual point source. If Cn is 1.0 or greater, then the cosine distribution will result (no matter how the source distance or Gx or Gy are set). If Cn is zero, but either Gx and Gy is non-zero, then a Gaussian distribution will result.

The parameters used to define this source are:

1-5: See "Parameters common to all source objects" .

6: X Half Width: The x half width in lens units.

7: Y Half Width: The y half width in lens units.

8: Source Distance: The distance along the local z axis from the apparent source point to the location of the source object. This value may be positive or negative. If zero, the rays are collimated. If positive, the apparent source point is behind the object. Considered only if Cn, Gx, and Gy are all zero.

9: Cosine Exponent: The power on the cosine term. This is Cn in the cosine distribution expression above. The ray cosines do not support Sobol sampling when this parameter is not zero.

10: Gaussian Gx: The X term in the Gaussian distribution. Ignored if Cn is not zero. The ray cosines do not support Sobol sampling when this parameter is not zero.

11: Gaussian Gy: The Y term in the Gaussian distribution. Ignored if Cn is not zero. The ray cosines do not support Sobol sampling when this parameter is not zero.

12: Source X: The X coordinate of the point that emits the rays. If Source Distance is zero, then the Source X parameter is the X direction cosine of the collimated ray bundle. Considered only if Cn, Gx, and Gy are all zero.

13: Source Y: The Y coordinate of the point that emits the rays. If Source Distance is zero, then the Source Y parameter is the Y direction cosine of the collimated ray bundle. Considered only if Cn, Gx, and Gy are all zero.

14: Min X Half Width: The minimum x half width in lens units. Use a value greater than zero and less than the X Half Width to define an annular region. The ray coordinates do not support Sobol sampling when this parameter is not zero.

15: Min Y Half Width: The minimum y half width in lens units. Use a value greater than zero and less than the Y Half Width to define an annular region. The ray coordinates do not support Sobol sampling when this parameter is not zero.

Use with LightningTrace

If the source is modeled as an imaging point source, then the X and Y Half Widths (parameters 6, 7) and the Source X and Source Y values (parameters 12, 13) contribute to the calculation of the angular distribution for the source; otherwise these parameters have no impact on the LightningTrace results. The size and spatial distribution of the source is neglected, and thus the Min Half Widths in X and Y (parameters 14, 15) are treated as zero by LightningTrace. Note that collimated input beams cannot be modeled by LightningTrace; Lightning- Trace will return zero flux for the beam if the Source Distance (parameter 8), Cosine Exponent (parameter 9), and Gaussian terms in X and Y (parameters 10, 11) are all zero.

Source EULUMDAT File

This source is defined by measured photometry data for a real lamp, properly formatted in a file commonly referred to as EULUMDAT format. The file extension must be LDT and the file must be placed in the <data>\Objects\Sources\EULUMDAT folder (see "Folders"). Although LDT files usually contain a specification of the total flux of the lamp, to remain consistent with other Zemax source models the total lamp flux is a user definable parameter.

The parameters are:

1-5: See "Parameters common to all source objects" .

Use with LightningTrace

This source type is fully supported by LightningTrace.

Source Filament

The Source Filament can be thought of as a thin wire coiled in a helix shape. The wire turns "N" times along the full Z coordinate length given by "L". The radius of the turns is defined by "R". Rays emanate from a randomly chosen point along the helix in a random direction.

The parameters used to define this source are:

1-5: See "Parameters common to all source objects" .

6: Length "L" in lens units.

7: Radius "R" in lens units.

8: Number of turns "N" (dimensionless). N may be fractional or even negative to reverse the rotation direction of the helix.

The axis of the helix, and the length L, are oriented along the Z axis.

Use with LightningTrace

The angular distribution of the source is uniform, i.e. it is identical to that for a point source. The size and spatial distribution of the source is neglected. As such, the Radius (parameter 7) and Number of turns (parameter 8) values have no impact on the LightningTrace results, and the only effect that the Length of the filament (parameter 6) has on the results is to position the center of the LightningTrace mesh at the axial center of the filament.

Source File

The Source File is a source whose ray coordinates, cosines, and intensity are defined in a user supplied file. The name of the file containing the ray data must be placed in the comment column of the object. The file extension may be either DAT or SDF and the file must be placed in the <data>\Objects\Sources\Source Files folder (see "Folders"). The extension SDF is recommended and DAT is supported only for backward compatibility. The file format may be either text or binary, both formats are described below.

The parameters are:

1-5: See "Parameters common to all source objects" .

6: Randomize?: If set to zero, the rays will normally be traced in the order listed in the file. If non-zero, then the ray order is randomized once, when the file is read or any parameter in the NSC Editor changes for the source object and the source is updated. The randomize feature is only available if the total number of rays in the file is less than the maximum number of source rays to hold in memory. Files with more than the maximum number of source rays to hold in memory are too large to hold in memory and cannot be randomized. These large files are partially read as required as described below. The maximum number of source rays to hold in memory is a user definable parameter; see "Maximum Source File Rays In Memory" .

For parameters 7, 8, and 9, the data presented is based only open the number of analysis rays read from the source file, which may be less than the total number of rays in the file.

7: Power/Lumens In File: The value for this parameter is for reference, and is set by Zemax upon reading the file and should not be altered or set by the user. The behavior of this field is different for "flux only" and "spectral color" formats, as defined in "Binary Source File format" .

For flux only format files:

This reference value is the total power in current source units defined by the rays in the file. Only the number of rays used for analysis is considered for making this power computation. If the current source units are photometric (lumens) then the power in watts in the source file are converted to photometric units at the monochromatic wavelength defined for the source file in the parameter 4 value. Note there is no way to convert power in watts to lumens if the source is polychromatic, since the wavelengths are randomly assigned as described in "Parameters common to all source objects" . If the source is polychromatic, this parameter will be set to zero. The actual power of each ray is defined by parameter 3 above, the number of rays being traced, and the relative power of each ray as defined in the source file.

For spectral color format files:

The wavelength of each ray is defined in the file, and those wavelengths will be used for tracing no matter what the wavelength parameter in the NSC Editor is set to. The Power/Lumens value will be valid even if this source is polychromatic, and the actual flux and wavelengths defined in the file are used to determine the total power in current source units.

8/9: These values are for reference only, and are set by Zemax upon reading the file and should not be altered or set by the user. These references values are the minimum and maximum wavelength in micrometers for the source. For spectral color format files, these values are read from the file. For flux only format files these values are determined by the system wavelengths or color spectrum defined for the source, if any. See "Defining the color and spectral content of sources" .

Text vs. binary format files

The maximum number of rays currently allowed for text files is 1,000,000. If there are more than 1,000,000 rays defined in a text source file, Zemax will offer to convert the file to binary format automatically. The original file will be renamed with the ".OLD" extension, and the new binary SDF file will be created using the same name as the original file. After the conversion is complete, an option will be presented to save or delete the old file. This conversion is done to save disk space (binary files are about 30% as large as equivalent text files), load time

(binary files read about 20x faster than text files), and system memory (large binary files can be left on disk and thus require very little RAM) even if billions of rays are being traced.

Memory requirements for source file objects

For both text and binary files, if the total number of rays to be traced is less than the maximum number of source rays in memory, Zemax stores the rays in system memory for maximum speed. This requires about 32 bytes of memory for each

ray, or up to 32 Mb of memory for 1,000,000 rays. If the total number of rays is greater than the maximum number of source rays in memory, Zemax leaves the ray data on disk, and reads the file as required. This requires less memory, and allows Zemax to trace a huge number of rays, limited only by the systems file storage capacity. The maximum number of source rays to hold in memory is a user definable parameter; see "Maximum Source File Rays In Memory" .

<u>Restrictions on the number of rays selected</u>

There are restrictions on the values for the number of layout and analysis rays when using the Source File:

- The number of analysis rays must be equal to or larger than the number of layout rays. The number of analysis rays will be set to the larger of the two numbers entered in the NSC Editor.

- The number of analysis and layout rays may not exceed the number of rays defined in the file.

<u>Binary Source File format</u>

The Binary Source File consists of a header structure of the form:

```
typedef struct
{
int Identifier; // Format version ID, current value is 1010
unsigned int NbrRays; // The number of rays in the file
char Description[100]; // A text description of the source
float SourceFlux; // The total flux in watts of this source
float RaySetFlux; // The flux in watts represented by this Ray Set
float Wavelength; // The wavelength in micrometers, 0 if a composite
float InclinationBeg, InclinationEnd; // Angular range for ray set (Degrees)
float AzimuthBeg, AzimuthEnd; // Angular range for ray set (Degrees)
long DimensionUnits; // METERS=0, IN=1, CM=2, FEET=3, MM=4
float LocX, LocY,LocZ; // Coordinate Translation of the source
float RotX,RotY,RotZ; // Source rotation (Radians)
float ScaleX, ScaleY, ScaleZ; // Currently unused
float unused1, unused2, unused3, unused4;
int ray_format_type, flux_type;
int reserved1, reserved2;
} NSC_RAY_DATA_HEADER;
```

The data types float, int, and unsigned int are all 32 bit types. Identifier defines the file format, currently 1010. Older formats normally can be read by Zemax as well, although Zemax only writes files in the current format.Zemax only uses the `NbrRays, DimensionUnits, ray_format_type,` and `flux_type` parameters. The other data is ignored.

The maximum valid number of rays in the file as defined by NbrRays is 4 billion (4,000,000,000 rays). Exceeding this number may cause problems with the reading of the data file by Zemax. The ray_format_type must be either 0 for flux only format, or 2 for the spectral color format (both are defined below). Other format types are not supported. If and only if the ray_format_type is 0, then the flux_type is 0 for watts, and 1 for lumens. For the spectral color format, the flux must be in watts, and the wavelength in micrometers. The flux and wavelength values and units are used in the ray structures defined below.

Note that when the spectral color format is used, Zemax will always trace rays at the wavelengths defined in the file, no matter what system wavelengths are defined

After the header follow `NbrRays` ray structures. The ray structure format depends upon the ray format type. For the flux only format, the ray structure is:

```
typedef struct
{
float x, y, z;
float l, m, n;
float flux;
} FLUX_ONLY;
```

For the color spectral color format, the ray structure is:

```
typedef struct
{
float x, y, z;
float l, m, n;
float flux, wavelength;
} SPECTRAL_COLOR;
```

Zemax can read either the flux only or the spectral color format. The spectral color format only is used by Zemax when generating source files from ray trace data (see the discussion of "Save Rays on Object n" in the section "The Ray Database Viewer").

Text Source File format

The text Source File consists of a single line of header data with just two integer numbers of the form:

`number_of_rays dimension_flag`

The number_of_rays indicates the total number of rays in the file. The dimension_flag is 0 for meters, 1 for inches, 2 for centimeters, 3 for feet, and 4 for millimeters.

The remaining lines in the file are of the format:

`x y z l m n intensity wavelength`

Any line starting with the "!" symbol is assumed to be a comment line and is ignored. Any number of rays may be defined in a single file using this text format. However, if the number of rays exceeds 1,000,000, the file will automatically be converted to binary format by Zemax the first time the file is opened for a source file object. For a discussion of this conversion see "Text vs. binary format files" . The wavelength argument is optional, however, if present, the ray will always be traced at the defined wavelength in micrometers, and the source file will be treated as described by the spectral color format in the discussion on binary file formats.

Each ray may have a different relative intensity. If the relative intensity value is not 1.0 for each ray, then normalization occurs as described below.

When the source file is first loaded into memory, the intensity of each ray is summed and then normalized to the average intensity. If the total flux of the source is then defined to be some number of watts, a subset of the rays can be traced and their intensities will yield approximately, but not exactly, the total flux. Normalization is required for an arbitrary subset of the rays to yield approximately the total desired power.

Use with LightningTrace

The angular distribution of the source is determined by creating a far-field map for the source upon first reading of the file. This map is stored in an FFD file located in the same folder as the source file. The map is created using all rays in the source, and thus the Randomize? value (parameter 6) has no effect on the LightningTrace results. The size and spatial distribution of the source is neglected.

Source Gaussian

The Source Gaussian has a Gaussian distribution of rays which appear to emanate from a point source. The parameters are:

1-5: See "Parameters common to all source objects" .

6: Beam Size: The beam radius at the 1 over e^2 point in intensity in lens units.

7: Position: The distance from the apparent point of divergence of the rays to the source plane location. If zero, the rays are collimated.

Use with LightningTrace

The angular distribution of the source is determined using the Beam Size (parameter 6) and Position (parameter 7) to calculate the far-field divergence angle of the source. However, the physical size of the beam as given by the Beam Size is not used, as the size and spatial distribution of the source is neglected.

Source IESNA File

This source is defined by measured photometry data for a real lamp, properly formatted in a file according to the Illumination Engineering Society of North America (IESNA) Standard File Format for Electronic Transfer of Photometric Data (IESNA91, IESNA LM-63-95, or IESNA LM-63-02). The file extension must be IES and the file must be placed in the <objects>\Sources\IESNA folder (see "Folders"). Although IES files usually contain a specification of the total flux of the lamp, to remain consistent with other Zemax source models the total lamp flux is a user definable parameter.

The parameters are:

1-5: See "Parameters common to all source objects".

Use with LightningTrace

This source type is fully supported by LightningTrace.

Source Imported

This source has a shape defined by an imported data file, in either IGES, STEP, or SAT format. The source volume has no properties other than to define the starting location and orientation of the rays. Rays are spatially distributed uniformly in area over the object. The angular distribution of the rays is given by

$$P(\theta) = (\cos\theta)^x$$

where θ is the angle measured from the local surface normal pointing outside the volume, and x is a user defined parameter between 0.0 and 400.0. If x = 0.0, the rays emanate in a hemisphere with equal probability in all directions. If x = 1.0, the distribution is Lambertian. If x >= 400.0, the rays are always normal to the surface.

The parameters are:

1-5: See "Parameters common to all source objects" .

6: Scale. This dimensionless parameter scales the entire imported solid. Upon import, Zemax will automatically attempt to scale the dimensions of the imported solid to match the current dimensions in Zemax; this scale factor is applied after that conversion.

7: Unused.

8: Cosine Factor: The cosine power exponent "x" described above.

Use with LightningTrace

This source type is not supported by LightningTrace.

Source Object

The source object creates a source with a size and shape based upon any other "parent" object. Any prior object in the NSC Editor may be used to define the shape of the Source Object, including User Defined and Boolean objects. Any changes in the properties of the parent object will dynamically affect the distribution of the rays on the Source Object. The parent object shape is used to determine the starting location and orientation of the rays. Rays are spatially distributed uniformly in area over the object. The angular distribution of the rays is given by

$$P(\theta) = (\cos\theta)^x$$

where θ is the angle measured from the local surface normal pointing outside the volume, and x is a user

defined parameter between 0.0 and 400.0. If x = 0.0, the rays emanate in a hemisphere with equal probability in all directions. If x = 1.0, the distribution is Lambertian. If x >= 400.0, the rays are always normal to the surface.

For best results, the parent object should be a solid volume. If the parent object is a surface, the rays may radiate from either one or both sides of the surface, depending upon exactly how the parent object is defined.

There are two ways to prevent rays leaving the source from immediately striking the parent object. The first way is to set the Source Object to ignore the parent object from which it is defined. This will allow the ray to ignore the parent object until after the ray leaves the source (See "Defining an Ignore Objects list"). The second way is to define a pre-propagation distance that is large compared to the chord tolerance but small compared to the distance the ray might legitimately travel before it interacts with the source itself (See "Sources tab"). This latter method is useful if the

interaction of the rays with the source body is desired and the source is hollow or concave so that rays leaving one part of the object may strike another part before striking any other object.

The Source Object may be placed independently from the parent object. To superimpose the Source Object on the parent object, set the reference object of the Source Object to the parent object (see "Reference objects") and leave the position and tilt values at zero.

The parameters are:

1-5: See "Parameters common to all source objects" .

6: Parent Object #. The integer object number of the object used to define the source shape. This object number must precede the object number of the Source Object in the NSC Editor.

7: Chord Tolerance: The chord tolerance determines the position accuracy of the starting point for the rays, and affects the rendering of the source. To render or trace rays from the source, Zemax converts the imported file to a list of triangles which approximate the shape. The tolerance is the maximum allowed distance in lens units between a single triangle and the actual surface of the source. More triangles are added if the tolerance is set smaller which yields more accurate rendering, at the expense of speed and a larger memory requirement. The default value of zero will use a chord tolerance related to the size of the object sufficient for most purposes.

8: Cosine Factor: The cosine power exponent "x" described above.

Use with LightningTrace

This source type is not supported by LightningTrace.

Source Point

The Source Point is a point which emits rays into a cone. The cone angle may be any value between 0 and

180 degrees (which would radiate into a full sphere). The parameters are:

1-5: See "Parameters common to all source objects" .

6: Cone angle: The semi-cone angle in degrees.

Use with LightningTrace

This source type is fully supported by LightningTrace.

Source Radial

The Source Radial is a flat, rectangular or elliptical shape that emits rays into a hemisphere. The distribution of the rays in this hemisphere is symmetric about the local z axis and is defined by a cubic spline fit to arbitrary intensity vs. angle data. The number of points can be any integer between 5 and 181, inclusive. The angular range is evenly divided among the specified number of points. The data for each point is the relative intensity measured

in the far field of the source at the angle corresponding to that point. The provided data is fit to a cubic spline. Rays generated from the spline fit will follow the correct distribution in the far field of the source.

The parameters are:

1-5: See "Parameters common to all source objects" .

6: X Half Width: The x half width in lens units. If less than zero, the emitting region will be elliptical.

7: Y Half Width: The y half width in lens units. If less than zero, the emitting region will be elliptical.

8: Minimum Angle in degrees. The value must be between 0.0 and 89.9.

9: Maximum Angle in degrees. The value must be between the minimum angle plus 0.1 and 90.0.

10: The number of points, must be between 5 and 181, inclusive.

11+: The relative intensity data at each angle from the normal.

Use with LightningTrace

The angular distribution of the source is taken for the input data (parameters 8, 9, 10, 11+). The size and spatial distribution of the source is neglected, and thus the X and Y Half Widths (parameters 6, 7) have no effect on the LightningTrace results.

Source Ray

The Source Ray is a point which emits rays along specified direction cosines. This is useful for debugging, for example, the direction cosines may be used to recreate a ray that causes geometry errors. The parameters are:

1-5: See "Parameters common to all source objects" .

6-8: Cosines: The local X, Y, and Z direction cosines of the ray. These are automatically normalized.

9: Random Seed: If not zero, the random number generator is reseeded with the specified value. This allows identical rays to be generated which subsequently have random but reproducible properties, such as when bulk or surface scattering is on.

Use with LightningTrace

This source type is not supported by LightningTrace.

Source Rectangle

The Source Rectangle is a flat rectangular surface which emits rays from a virtual source point. The parameters are identical to the Source Ellipse, but the shape of the source is a rectangle rather than an ellipse.

Use with LightningTrace

See the description under the Source Ellipse.

Source Tube

The Source Tube is similar to the Source Volume Cylinder, except the rays only emanate from the surface of the tube rather than the full volume.

The parameters used to define this source are:

1-5: See "Parameters common to all source objects" .

6: Length "L" in lens units.

7: Radius "R" in lens units.

The axis of the tube, and the length L, are oriented along the Z axis.

The local origin of the Source Tube is at the center of the front face of the tube.

Use with LightningTrace

The angular distribution of the source is uniform, i.e. it is identical to that for a point source. The size and spatial distribution of the source is neglected. As such, the Radius value (parameter 7) has no impact on the LightningTrace results, and the only effect that the Length of the tube (parameter 6) has on the results is to position the center of the LightningTrace mesh at the axial center of the tube.

Source Two Angle

The Source Two Angle is a flat surface, with either an elliptical or a rectangular shape, which emits rays. The origin of each ray launched lies on the surface of the source with a uniform distribution. The angular distribution of the rays is either uniform in angle space or uniform as projected onto a distant plane. The angular distribution may also be either rectangular or elliptical, with distinct maximum half angles for each direction.

Using just two orthogonal angles to describe a source distribution under-constrains the possible energy distribution, since there is no clear definition of what happens at intermediate azimuthal angles. For this reason, this source model should only be used at angles less than roughly 60 degrees, and only when the aspect ratio between the X and Y angles is no more than about 4 to 1. It is a good idea to test the resulting energy distribution using a polar detector to confirm the approximations used in this model yield the desired distribution.

The parameters used to define this source are:

1-5: See "Parameters common to all source objects" .

6: X Half Width: The x half width in lens units.

7: Y Half Width: The y half width in lens units.

8: X Half Angle in degrees: The half angle of the cone of rays in the XZ plane. Maximum 89 degrees.

9: Y Half Angle in degrees: The half angle of the cone of rays in the YZ plane. Maximum 89 degrees.

10: Spatial Shape: Use 0 for rectangular, 1 for elliptical.

11: Angular Shape: Use 0 for rectangular, 1 for elliptical.

12: Uniform Angle: Use 0 for uniform irradiance on a distant projected plane, or 1 for uniform in angle space.

Use with LightningTrace

The angular distribution of the source is determined from the X and Y Half Angle values (parameters 8, 9), the Angular Shape value (parameter 11), and the Uniform Angle value (parameter 12). The size and spatial distribution of the source is neglected. As such, the X and Y Half Width values (parameters 6, 7) and the Spatial Shape value (parameter 10) have no impact on the LightningTrace results.

Source Volume Cylinder

The Source Volume Cylinder is a 3D volume formed by an elliptical shape in the XY plane that is symmetrically extruded along the Z axis. The center of the source volume is at the local origin of the object. Rays are emitted in random directions from anywhere inside the volume, with uniform probability in both position and ray direction. Points within the cylinder volume satisfy these relations:

$$\left(\frac{X}{W_x}\right)^2 + \left(\frac{Y}{W_y}\right)^2 \leq 1 \text{, and } \left(\frac{Z}{W_z}\right)^2 \leq 1$$

where W refers to the half widths in X, Y, and Z. The parameters are:

1-5: See "Parameters common to all source objects" .

6: X Half Width: The x half width in lens units.

7: Y Half Width: The y half width in lens units.

8: Z Half Width The z half width in lens units.

Use with LightningTrace

The angular distribution of the source is uniform, i.e. it is identical to that for a point source. The size and spatial distribution of the source is neglected. As such, the X, Y, and Z Half Width values (parameters 6-8) have no effect on the LightningTrace results, other than to position the center of the LightningTrace mesh at the center of the cylindrical volume.

Source Volume Rectangle

The Source Volume Rectangle is a 3D volume formed by a rectangular shape in the XY, XZ, and YZ planes. The center of the source volume is at the local origin of the object. Rays are emitted in random directions from anywhere inside the volume, with uniform probability in both position and ray direction. Points within the rectangle satisfy these relations:

$$|X| \leq W_x, \ |Y| \leq W_y, \text{ and } |Z| \leq W_z$$

where W refers to the half widths in X, Y, and Z. The parameters are otherwise identical to that for the Source Volume Cylinder.

Use with LightningTrace

The angular distribution of the source is uniform, i.e. it is identical to that for a point source. The size and spatial distribution of the source is neglected. As such, the X, Y, and Z Half Width values (parameters 6-8) have no effect on the LightningTrace results, other than to position the center of the LightningTrace mesh at the center of the rectangular

Non-Sequential Detectors

The available types of detectors in Zemax are described in this section.

Summary of NSC Dectectors

Object Name	Description
Detector Color	A flat rectangular detector with an arbitrary number of pixels. This detector can record and display incoherent illumination data defined by tristimulus response. This detector can accurately record and display the color of illumination
Detector Polar	A section of a sphere, or a full sphere, that collects angular (far field) intensity data. Data collected by this detector may be exported into source data files such as IESNA and EULUMDAT.
Detector Rectangle	A flat rectangular detector with an arbitrary number of pixels. This detector can record and display incoherent, coherent, point spread function, polarization, and other data. This is the most powerful detector in terms of the analysis it provides, but it is limited to a flat rectangular shape.
Detector Surfce	A circular or annular detector with an arbitrary number of pixels in the radial and angular directions. The surface may follow a plane, sphere, conic asphere, or aspherical shape. The surface detector can only record incoherent irradiance data.
Detector Volume	A rectangular volume with an arbitrary number of voxels in the local x, y, and z directions. The detector volume may be nested within or straddle any other object. Multiple detector volumes may also be superimposed and all will be illuminated by rays passing through the individual voxels.
Objects as detectors	Most objects of arbitrary shape may be used as a detector that records incoherent irradiance data.

For more information, see the following detailed descriptions of these detector types.

Detector Color object
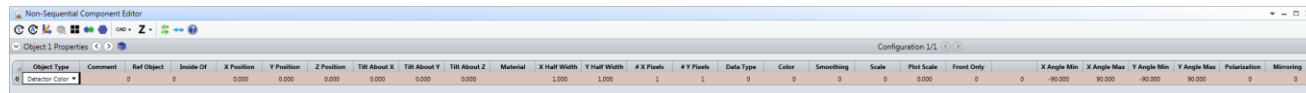
The Detector Color object stores power and tristimulus (visual light color) data from NSC source rays that strike it. The resulting data distributions may be viewed for incoherent light in spatial or angular domains. This object may optionally store the spectral distribution for all NSC source rays that strike it (see the description for "Record spectral data" given under the "Type tab"). Detector Color objects may be reflective, transparent, or absorbing if the material is set to "MIRROR", blank, or "ABSORB", respectively.

The defining parameters are:

1: X Half Width. The x width in lens units.

2: Y Half Width. The y width in lens units.

3: # X Pixels. The number of pixels along the x direction (see discussion below).

4: # Y Pixels. The number of pixels along the y direction (see discussion below).

Parameters 5-9 are only used to define how the detector is displayed on shaded model plots:

5: Data type: Select 0 for irradiance, 1 for intensity. If photometric units are selected, the options are the equivalent photometric quantities. For more information on selecting units, see "Analysis Units" .

6: Color: Select for 0 grey scale, 1 for inverse grey scale, 2 for false color, 3 for inverse false color, and 4 for true color.

7: Smoothing: The amount of smoothing. The algorithm is described in "The Detector Viewer" .

8: Scale: Choose 0 for linear, 1 for log -5, 2 for log -10, and 3 for log -15. This option is ignored if Color is set to 4 for true color.

9: Plot Scale: choose a maximum value to normalize the color display to; this is useful for setting a common scale across multiple detectors. This value is ignored if it does not exceed the maximum displayed value.

10: Front Only. If 0, then rays may strike the detector on the front or the back side. If this flag is 1, then rays coming from the back will be ignored, and will miss the detector. The "back" side is the side facing towards the positive local z axis.

11: Unused.

12-15: The minimum and maximum x and y direction angles in degrees. These settings are used to control the sensitivity of the detector to rays incident upon the detector at an angle when displaying intensity. The default setting is -90.0 degrees to +90.0 degrees in both x and y directions, which allows display of angle data for all rays striking the detector. If the angular range is set to cover a subset of the possible incident angles, rays that strike the detector at angles outside the defined range are ignored ONLY for intensity displays.

16: Polarization flag. If 0, the detector ignores polarization of the incident rays and considers all incident rays. If the flag is 1, 2, or 3, the detector will only consider the component of the rays polarized along the local X, Y, and Z directions, respectively, and will consider the phase of the field along those directions as part of the optical path length. If the flag is 4, 5, or 6, the detector will only consider the component of the rays polarized along X and Y, X and Z, and Y and Z, respectively, and ignore the phase of the field. If the flag is 7, 8, or 9, the detector will only consider the component of the rays polarized along the local X, Y, and Z directions, respectively, and will ignore the phase of the field along those directions.

17: Mirroring flag. Mirroring allows the detector to exploit symmetry in the incident rays. Any ray that lands on the detector will be divided into parts, and each part of the ray will be copied to the symmetric mirror images of the

detector. If mirroring is set to 0, the detector does not use the mirroring feature. If mirroring is set to 1, then the detector will assume left-right symmetry exists in the incident rays. If mirroring is set to 2, then the detector will assume top-bottom symmetry. If mirroring is set to 3, then both left-right and top-bottom mirroring will be done. The mirroring may also be set to 4, 5, or 6; these values are similar to 1, 2, and 3, respectively, except the intensity of the rays is not scaled down to conserve energy. For example, using a value of 1 would divide a single 1-watt ray into two 0.5 watt rays, while a value of 4 would create two 1-watt rays. Which value is correct depends upon whether or not the sources generating the illumination are also scaled down to reflect symmetry or not, and only the user can make this determination. Note that if mirroring is used, and the illumination and optics are not actually symmetric in the expected way, erroneous data may result.

The information stored on any detector may be viewed by selecting Detectors, Detector Viewer on the NSC Editor, or when the program mode is Non-sequential, on the Analysis menu. The Detector Viewer also has an option for displaying data from any previously saved ray data base. If "none" is selected for the ray database, then the data currently stored in the detector will be displayed, otherwise, the data stored in the selected ray database will be used to regenerate the displayed image. For more information see "The Detector Viewer" . The maximum number of either X or Y pixels is 5000 for 32-bit versions of Zemax, and 6000 for 64-bit versions of Zemax.

For more information on the definitions and units for the terms irradiance, intensity, and flux, see "Analysis Units".

Comments on Detector Color pixel numbering

For the Detector Color, the pixel numbers start at 1 in the lower left (-x, -y) corner of the rectangle. The pixel numbers increase along the +x direction first, and then move up one row in the +y direction, and start over at the

-x edge. For a detector with nx by ny pixels, the pixel numbers are 1 through nx on the bottom row, and then nx+1

through 2nx on the next row above the bottom, until the last pixel (number nx*ny) is reached in the upper right (+x, +y) corner. Another way of stating the pixel numbering is that the x index changes the fastest, and then the y index.

Comments on Pixel Interpolation

See "Pixel Interpolation" under "Type tab".

Use with LightningTrace

LightningTrace may be used to view the irradiance distribution on this detector, but not the intensity distribution. When using LightningTrace, the intensity distribution may be obtained only via the Detector Polar object (see "Detector Polar object"). LightningTrace does not account for polarization (thus parameter 16, the Polarization flag, is not used) and cannot be used to exploit symmetry in the incident rays (thus parameter 17, the Mirroring flag, is not used). The interpolation algorithm used by LightningTrace does not depend on the option to "Use Pixel Interpolation" found under the Type tab of the Object Properties dialog box, and thus this option has no effect on the LightningTrace results.

Detector Polar object
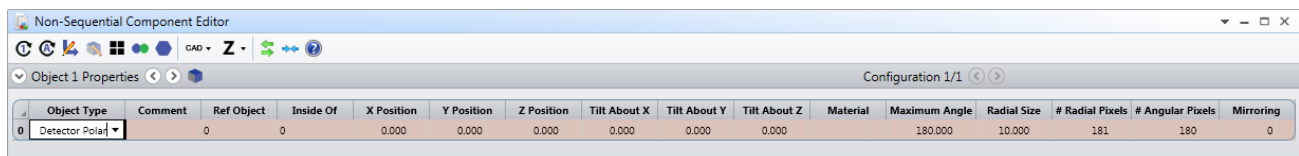
The Detector Polar object stores power and tristimulus (visual light color) data from NSC source rays that strike it. The resulting data distributions may be viewed for incoherent light in the angular domains. Detector Polar objects may be reflective, transparent, or absorbing if the material is set to "MIRROR", blank, or "ABSORB", respectively.

The object shape is a faceted approximation to a portion of a sphere centered on the object coordinates. Although the detector only records angular data, the detector must be large enough to encompass all the rays of interest, and the size of the detector is a user-definable parameter. The polar object spans an angular region, with the zero polar angle centered around the object local Z axis. The polar angle may span any angle between 1 and

180 degrees (which would form a complete sphere). The detector only considers the angle the ray striking the

detector makes, and not the position the ray strikes the detector. The azimuthal angle is defined by

$$\phi = \tan^{-1}\left(\frac{y}{x}\right)$$

where the zero angle is along the +x axis and the angle increases in a counter-clockwise direction all the way to 360 degrees. The azimuthal angle always spans 360 degrees. The defining parameters are:

1: Maximum Angle. The maximum polar angle in degrees. Must be between 1 and 180 degrees.

2: Radial Size. The radial size of the detector in lens units.

3: # Radial Pixels. The number of pixels along the polar angle direction. Must be between 10 and 721.

4: # Angular Pixels. The number of pixels along the azimuth angle direction. Must be between 12 and 720, and must be a multiple of 4.

5: Mirroring flag. Mirroring allows the detector to exploit symmetry in the incident rays. Any ray that lands on the detector will be divided into parts, and each part of the ray will be copied to the symmetric mirror images of the detector. If mirroring is set to 0, the detector does not use the mirroring feature. If mirroring is set to 1, then the detector will assume left-right symmetry exists in the incident rays. If mirroring is set to 2, then the detector will assume top-bottom symmetry. If mirroring is set to 3, then both left-right and top-bottom mirroring will be done. The mirroring may also be set to 4, 5, or 6; these values are similar to 1, 2, and 3, respectively, except the intensity of the rays is not scaled down to conserve energy. For example, using a value of 1 would divide a single 1-watt ray into two 0.5 watt rays, while a value of 4 would create two 1-watt rays. Which value is correct depends upon whether or not the sources generating the illumination are also scaled down to reflect symmetry or not, and only the user can make this determination. Note that if mirroring is used, and the illumination and optics are not actually symmetric in the expected way, erroneous data may result.

The information stored on any detector may be viewed by selecting Detectors, Detector Viewer on the NSC Editor, or when the program mode is Non-sequential, on the Analysis menu. The Detector Viewer also has an option for displaying data from any previously saved ray data base. If "none" is selected for the ray database, then the data currently stored in the detector will be displayed, otherwise, the data stored in the selected ray database will be used to regenerate the displayed image. For more information see "The Detector Viewer" .

For more information on the definitions and units for the terms irradiance, intensity, and flux, see "Analysis Units".

Exporting Detector Polar Data

Intensity data stored on a Detector Polar maybe exported to a source file. See "Export Polar Detector Data as Source File" .

Comments on Detector Polar Pixel Numbering

For the Detector Polar, the pixel numbers start at 1 in the polar center at an azimuthal angle of zero. The pixel numbers increase along the polar direction until pixel np, where np is the number of polar pixels. The pixel np+1 is at zero polar angle at the first azimuthal increment counter-clockwise from 0 degrees, and the pixel number increase along this azimuthal arm until 2*np. The next pixel number is 2*np+1, which starts again at a polar angle of zero and proceeds

along the next azimuthal arm. Another way of stating the pixel numbering is that the polar index changes the fastest, and then the azimuthal index.

Comments on Pixel Interpolation

See "Pixel Interpolation" under "Type tab" .

Use with LightningTrace

LightningTrace cannot be used to exploit symmetry in the incident rays, and thus parameter 5, the Mirroring flag, is not used. The interpolation algorithm used by LightningTrace does not depend on the option to "Use Pixel Interpolation" found under the Type tab of the Object Properties dialog box, and thus this option has no effect on the LightningTrace results. Finally, to avoid aliasing, any intensity data near the 180 degree radial boundary of the detector is neglected when the LightningTrace results are used to illuminate this detector.

Detector Rectangle object

The detector rectangle object stores energy data from NSC source rays that strike it. The resulting data distributions may be viewed for incoherent light in spatial or angular domains, or spatially coherent irradiance or phase, or coherent irradiance as a point spread function. Detector rectangles may be reflective, transparent, or absorbing if the material is set to "MIRROR", blank, or "ABSORB", respectively.

The defining parameters are:

1: X Half Width. The x width in lens units.

2: Y Half Width. The y width in lens units.

3: # X Pixels. The number of pixels along the x direction (see discussion below).

4: # Y Pixels. The number of pixels along the y direction (see discussion below).

Parameters 5-9 are only used to define how the detector is displayed on shaded model plots:

5: Data type: Select 0 for incoherent irradiance, 1 for coherent irradiance, 2 for coherent phase, 3 for radiant intensity, 4 for radiance in position space, and 5 for radiance in angle space. If photometric units are selected, the options are the equivalent photometric quantities. For more information on selecting units, see "Analysis Units" .

6: Color: Select for 0 grey scale, 1 for inverse grey scale, 2 for false color, and 3 for inverse false color.

7: Smoothing: The amount of smoothing. The algorithm is described in "The Detector Viewer" .

8: Scale: Choose 0 for linear, 1 for log -5, 2 for log -10, and 3 for log -15.

9: Plot Scale: choose a maximum value to normalize the color display to; this is useful for setting a common scale across multiple detectors.

10: Front Only. If 0, then rays may strike the detector on the front or the back side. If this flag is 1, then rays coming from the back will be ignored, and will miss the detector. The "back" side is the side facing towards the positive local z axis.

11: PSF Mode Wavenumber. If 0, then the coherent irradiance display is the coherent data detected for each pixel individually. This is the default and most appropriate choice if the detector is large and the intent is to look at the beam far from focus, such as fringes created by two overlapping beams from an interferometer. If the wavenumber is equal to a wavelength number, then the coherent data for each pixel is the Huygens integral of all rays incident

upon the detector, resulting in a diffraction Huygens Point Spread Function. When using PSF Mode, the width of the detector should be a small number of wavelengths, to allow adequate pixel

resolution to see the diffraction structure, and to avoid aliasing artifacts inherent to the Huygens model when applied to large areas. The irradiance in is normalized so that the total power is equal to the total incoherent power on the detector. This gives accurate results as long as the rays incident upon the detector cover a broad enough range of angles so that both constructive and destructive interference occur over the size of the detector. This normalizing will not give accurate results if the incident rays are all nearly parallel and PSF Mode should not be used for this purpose. If PSF Mode is used and the detector is placed inside of another object, such as a block of glass, then the "Inside Of" value for the detector should be used to indicate which object the detector is immersed within.

12-15: The minimum and maximum x and y direction angles in degrees. These settings are used to control the sensitivity of the detector to rays incident upon the detector at an angle when displaying radiant intensity or radiance in angle space. The default setting is -90.0 degrees to +90.0 degrees in both x and y directions, which allows display of angle data for all rays striking the detector. If the angular range is set to cover a subset of the possible incident angles, rays that strike the detector at angles outside the defined range are ignored ONLY for radiant intensity and radiance in angle space displays.

16: Polarization flag. If 0, the detector ignores polarization of the incident rays and considers all incident rays. If the flag is 1, 2, or 3, the detector will only consider the component of the rays polarized along the local X, Y, and Z directions, respectively, and will consider the phase of the field along those directions as part of the optical path length. If the flag is 4, 5, or 6, the detector will only consider the component of the rays polarized along X and Y, X and Z, and Y and Z, respectively, and ignore the phase of the field. If the flag is 7, 8, or 9, the detector will only consider the component of the rays polarized along the local X, Y, and Z directions, respectively, and will ignore the phase of the field along those directions.

17: Mirroring flag. Mirroring allows the detector to exploit symmetry in the incident rays. Any ray that lands on the detector will be divided into parts, and each part of the ray will be copied to the symmetric mirror images of the detector. If mirroring is set to 0, the detector does not use the mirroring feature. If mirroring is set to 1, then the detector will assume left-right symmetry exists in the incident rays. If mirroring is set to 2, then the detector will assume top-bottom symmetry. If mirroring is set to 3, then both left-right and top-bottom mirroring will be done. The mirroring may also be set to 4, 5, or 6; these values are similar to 1, 2, and 3, respectively, except the intensity of the rays is not scaled down to conserve energy. For example, using a value of 1 would divide a single 1-watt ray into two 0.5 watt rays, while a value of 4 would create two 1-watt rays. Which value is correct depends upon whether or not the sources generating the illumination are also scaled down to reflect symmetry or not, and only the user can make this determination. Note that if mirroring is used, and the illumination and optics are not actually symmetric in the expected way, erroneous data may result.

The information stored on any detector may be viewed by selecting Detectors, Detector Viewer on the NSC Editor, or when the program mode is Non-sequential, on the Analysis menu. The Detector Viewer also has an option for displaying data from any previously saved ray data base. If "none" is selected for the ray database, then the data currently stored in the detector will be displayed, otherwise, the data stored in the selected ray database will be used to regenerate the displayed image. For more information see "The Detector Viewer" . The maximum number of either X or Y pixels is 5000 for 32-bit versions of Zemax, and 6000 for 64-bit versions of Zemax.

Four data items are stored for every ray that strikes the detector:

The incoherent intensity of the ray. This energy is stored by incrementing the counter corresponding to the pixel the ray struck. The sum of the ray intensity divided by the pixel area yields the irradiance in flux per area.

The incoherent intensity of the ray in angular space. This energy is stored by incrementing the counter corresponding to the pixel the ray struck in angle space. The sum of the ray intensity divided by the solid angle the pixel represents yields the intensity in flux per solid angle.

The coherent real and imaginary amplitudes of the ray. The amplitude is the square root of the ray intensity, while the phase is determined by the total optical path length from the source referenced to the center of the spatially distributed pixel. By separating the real and imaginary parts of the ray, interference between many rays may be simulated. Phase due to polarization effects is also considered if the polarization flag is 1, 2, or 3; see the discussion for parameter 16 above.

For more information on the definitions and units for the terms irradiance, intensity, and flux, see "Analysis Units".

Comments on coherent data computations

The propagation and interference of light generally has properties of both particles and waves. Rays can be thought of as the particle representation, and diffraction interference (such as for a diffraction PSF) can be thought of as a wave representation.

For NSC analysis, Zemax uses ray tracing to determine optical paths and energy distributions. Zemax accounts for the phase along the ray, and this allows for computation of some interference and diffraction effects. However, it is important that the user understand what assumptions the model makes and how these assumptions affect the accuracy of the results.

When a ray strikes a detector, Zemax computes the real and imaginary parts of the electric field by using the intensity and phase of the ray referenced to the center of the pixel struck. The real and imaginary parts may then be summed for many rays that strike the same pixel. Zemax also sums the intensity (amplitude squared) for each pixel.

Because the phase is accounted for, some rays will constructively interfere with other rays while other rays will destructively interfere. This allows Zemax to simulate effects such as fringes in interferometers (shearing or otherwise) or interference from various orders of a diffraction grating. However, computing the coherent irradiance involves some assumptions. Physically, destructive interference means the energy would have propagated somewhere other than where the ray went. In a similar way, when constructive interference occurs, the squaring of the amplitude sum of many rays artificially and nonphysically increases the energy in the beam. Zemax cannot determine where the energy went (or came from), and therefore cannot account for conservation of energy in coherent irradiance calculations without making assumptions. Zemax can make one of two different assumptions: either to normalize the coherent power on the detector to match the incoherent power, or to determine the coherent power using the incoherent power on each pixel and scaling by the degree of coherence of rays landing on each pixel. Which method is used depends upon the number of pixels on the detector, and on the setting of the Normalize Coherent Power checkbox (see "The object properties dialog box").

If there is only one pixel defined for the detector, or if the Normalize Coherent Power checkbox is off, then the coherent irradiance for each pixel is computed by summing the real and imaginary parts of every ray incident upon that pixel, computing the magnitude of this sum, dividing by the square of the sum of the coherent amplitude of all incident rays upon that pixel, then finally multiplying this ratio by the incoherent irradiance on the pixel. This method allows the computed coherent irradiance to vary between a value of zero and the incoherent irradiance. However, there is no way to accurately determine the true coherent irradiance in this case because of the limitation of the ray model in the presence of constructive and destructive interference as described above. Specifically, it is unknowable where energy lost in this computation would have propagated to.

If there is more than one pixel in the detector, and the Normalize Coherent Power checkbox is on, then the coherent irradiance is computed from the ray data by squaring and summing the real and imaginary parts of the amplitude pixel by pixel, and then re-normalizing the total coherent irradiance for the entire detector to be equal to the incoherent irradiance incident upon the detector.

With two methods available, the question naturally arises - which method is correct? If one or the other method was always correct, there would be no need for the Normalize Coherent Power switch. The general rule is simple. If all the incoherent power from one or more sources falls on the same detector, then the Normalize Coherent Power switch should be checked on. This is the default setting. This would be case, for example, of two plane waves interfering to form a sinusoidal fringe pattern with many fringes across a detector. All of the energy is present on the detector, but the wavefront interference causes ray energy that would have landed in a dark fringe to be redirected to a neighboring bright fringe on the same detector. Conversely, when two parallel, perfectly collimated beams are combined, yielding a single bright or dark (or something in between) fringe, then the Normalize Coherent Power switch should be checked off. This would be the case where an interferometer was set up to yield a single dark fringe on one detector and a single bright fringe on another detector - physically the energy never would propagate along the dark path, but the rays cannot know this in advance. Note that in this case (even when using the recommended setting), the bright fringe will only show half the actual power - the rays cannot create power, and the coherent power computation is still capped at the incoherent energy. The energy lost in the dark fringe physically would have been directed to the bright fringe, but again, the rays are incapable of determining that.

There may be cases where neither of these coherent power computation methods yields the correct results. These cases are simply beyond the scope of the ray model, and Zemax should not be used for critical analysis of the physical optics aspect of these systems.

It is important to understand that Zemax considers ALL sources to be coherent with respect to one another, and for the phase of the ray to be zero at the starting coordinate of the ray, wherever it may be. This generally limits the usefulness of the interference analysis to monochromatic sources. Polychromatic sources cannot be modeled correctly for coherent analysis because all rays, regardless of their wavelength, are coherently summed. The initial phase of the ray and the coherence length of the source may be defined, see "Coherence length modeling" and "Sources tab" .

Comments on Detector Rectangle angular binning

For the purposes of binning energy into angle space pixels, Zemax computes the X and Y angles from polar coordinates. The radial angle of the ray is measured with respect to the surface normal, and is then projected onto the surface to compute the X and Y angles. Note that these angles are slightly different than the Cartesian XZ and YZ angles used in some other contexts, especially when using compound tilts. See "Field angles and heights" for an example of the Cartesian angle methodology.

Comments on Detector Rectangle pixel numbering

For the Detector Rectangle, the pixel numbers start at 1 in the lower left (-x, -y) corner of the rectangle. The pixel numbers increase along the +x direction first, and then move up one row in the +y direction, and start over at the -x edge. For a detector with nx by ny pixels, the pixel numbers are 1 through nx on the bottom row, and then nx+1 through 2nx on the next row above the bottom, until the last pixel (number nx*ny) is reached in the upper right (+x, +y) corner. Another way of stating the pixel numbering is that the x index changes the fastest, and then the y index.

Comments on Pixel Interpolation

See "Pixel Interpolation" under "Type tab" .

Use with LightningTrace

LightningTrace may only be used to view the incoherent irradiance distribution on this detector (as given by the "incoherent irradiance" and "radiance in position space" options for the detector data type), but not the coherent irradiance or the intensity. When using LightningTrace, the intensity distribution may be obtained only via the Detector Polar object (see "Detector Polar object"). LightningTrace does not account for polarization (thus parameter 16, the Polarization flag, is not used) and cannot be used to exploit symmetry in the incident rays (thus parameter 17, the Mirroring flag, is not used). The interpolation algorithm used by Lightning- Trace does not depend on the option to "Use Pixel Interpolation" found under the Type tab of the Object Properties dialog box (thus this option has no effect on the LightningTrace results).

Detector Surface object

The detector surface object stores energy data from NSC source rays that strike it. The resulting data distributions may be viewed as irradiance for incoherent light. Detector surfaces may be reflective, transparent, or absorbing if the material is set to "MIRROR", blank, or "ABSORB", respectively.

The surface shape of a detector surface object is identical to the Aspheric Surface object. See "Aspheric Surface" for information on the defining formulas for the surface shape.

The defining parameters are:

1: Radius: The radius of curvature in lens units. If zero, a plane base surface will result.

2: Conic: The conic constant. See "Aspheric Surface" for the surface shape formula.

3: Max Aperture: The maximum radial size in lens units.

4: Min Aperture: The minimum radial size in lens units. If this value is greater than zero, an annular detector will be created.

5: # A Zones: The number of angular zones, Na. See "Comments about detector triangles" below.

6: # R Zones: The number of radial zones, Nr. See "Comments about detector triangles" below.

Parameters 7-11 are only used to define the appearance of the Detector Surface on Shaded Model Plots:

7: Data type: Select 0 for incoherent irradiance, 1 for incoherent flux.

8: Color: Select for 0 grey scale, 1 for inverse grey scale, 2 for false color, and 3 for inverse false color.

9: Unused.

10: Scale: Choose 0 for linear, 1 for log -5, 2 for log -10, and 3 for log -15.

11: Max Scale: choose a maximum value to normalize the color display to; this is useful for setting a common scale across multiple detectors.

12: Front Only. If 0, then rays may strike the detector on the front or the back side. If this flag is 1, then rays coming from the back will be ignored, and will miss the detector. The "back" side is the side facing towards the positive local z axis.

13: The number of terms to use in the aspheric expansion. Ray tracing will be faster if this term is no larger than the highest order non-zero coefficient.

14-250: The α coefficients on the polynomial expansion. See "Aspheric Surface" for the surface shape formula.

The information stored on any detector may be viewed by selecting Detectors, Detector Viewer on the NSC Editor, or when the program mode is Non-sequential, on the Analysis menu. The Detector Viewer also has an option for displaying data from any previously saved ray data base. If "none" is selected for the ray database, then the data currently stored in the detector will be displayed, otherwise, the data stored in the selected ray database will be used to regenerate the displayed image. For more information see "The Detector Viewer" .

One data item is stored for every ray that strikes the detector: The incoherent intensity of the ray. This energy is stored by incrementing the counter corresponding to the pixel the ray struck. The sum of the ray intensity divided by the pixel area yields the irradiance in flux per area.

For more information on the definitions and units for the terms irradiance, intensity, and flux, see "Analysis Units".

Comments about detector triangles

Zemax uses a collection of triangles to conform to the shape of the curved surface. There are Nr radial zones, and Na angular zones. Together these two parameters define Nr * Na polygonal regions. Each region is generally divided into two triangles, so the total number of triangular pixels is 2 * Nr * Na. However, if the minimum aperture (parameter 4 above) is zero, then the innermost radial zone has only Na triangles rather than 2 * Na. In this case the total number of triangles is 2 * Nr * Na - Na. The algorithm that selects the triangle coordinates attempts to make the area of each triangle approximately the same, however the area from triangle to triangle can not be held exactly constant. The corners of each triangle will lie on the aspheric surface defined by the radius, conic, and aspheric coefficients. The triangles are flat and do not lie exactly on the surface if the surface is not flat.

The triangles are the individual pixels used to record irradiance and flux data. The pixels are numbered in consecutive radial zones. Pixel 1 starts at angle zero in the innermost radial zone. The first 2 * Na pixels (or just Na pixels if the minimum aperture is zero) are numbered going counter clockwise around z to complete the first annular zone. The pixel numbering continues at angle zero in the second radial zone for the next 2 * Na pixels, until all radial zones are complete.

If "Fast Ray Trace" (see "Type tab") is checked, Zemax will alter the order of the triangles to speed up the ray trace. If "Fast Ray Trace" is unchecked, the order of the triangles is unmodified, which slows the ray tracing but preserves the ordering described above.

Use with LightningTrace

This detector type is not supported by LightningTrace.

Detector Volume object
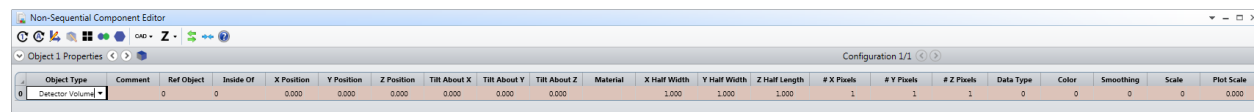
The detector volume object stores incident and absorbed energy data from NSC source rays that pass through the individual voxels. The resulting data distributions may be viewed as incident flux, absorbed flux, or absorbed flux per unit volume. Detector volumes may be transparent, or be made of any valid material. Detector volumes may also be nested within or straddle other objects. Objects within detector volumes may be refractive, reflective, absorbing, gradient index, or have surface and or bulk scatter properties defined.

The shape of a detector volume object is a rectangular solid. The defining parameters are:

1-3: The X, Y, and Z half widths of the volume. The center of the volume is at the local coordinate origin.

4-6: The number of voxels in the X, Y, and Z directions. The total number of voxels is nx * ny * nz.

The information stored on any detector may be viewed by clicking the Detector Viewer button in the Analysis tab when the program is in non-sequential or mixed UI mode. Data for detector volume objects is not available when using previously saved data from a ray data base. Pressing the left and right arrow keys while viewing detector volume data on the NSC detector viewer will cycle through all the Z-planes. For more information see "The Detector Viewer" .

Two data items are stored for every ray that passes through each voxel: The incident flux of the ray in source units, and the absorbed flux in the voxel. The incident flux is the flux of the ray as it enters the voxel. A ray segment that starts in a particular voxel will not increase the incident flux in that voxel, because the incident flux is only incremented for rays entering the voxel. The absorbed flux is a function of the path length of the ray through the voxel, as well as the coefficient of absorption of the material the ray is passing through. Note that the total absorbed flux only considers absorption in the object volume, and not at the object coating.

Detector volumes may be nested within other NSC object types. For example, to measure the absorbed flux in a cylindrical glass rod, place the rod within the detector volume, and set the glass type of the rod on the rod object. If the glass type has an absorption coefficient associated with the material, the portion of the flux of rays passing through the glass rod will be recorded by the detector volume.

For more information on the definitions and units for the terms irradiance, intensity, and flux, see "Analysis Units".

<u>Comments on Detector Volume pixel numbering</u>

For the Detector Volume, the pixel numbers start at 1 in the lower left corner of the rectangle formed by the first z-plane (on the -z side of the volume). The pixel numbers in this z-plane increase along the +x direction first, and then move up one row in the +y direction, and start over at the -x edge. For a detector with nx by ny by nz pixels, the pixel numbers in the first z-plane are 1 through nx on the bottom row, and then nx+1 through 2nx on the next row above the bottom, until the pixel nx*ny is reached in the upper right (+x, +y) corner. The numbering then proceeds to the next z-plane, starting again at the bottom left corner. Another way of stating the pixel numbering is that the x index changes the fastest, then the y index, and then finally the z index.

<u>Use with LightningTrace</u>

This detector type is not supported by LightningTrace.

Objects as detectors

Most objects may act like a detector. This includes objects with flat faces, such as the polygon, STL, and rectangular volume objects. Most objects that approximate the shape of a curved surface or solid for rendering purposes by using the drawing resolution setting (see "Draw tab"), such as the aspheric surface, toroidal lens, biconic lens, and other complex shapes, may also be detectors.

To make any of these objects a detector, choose "Object Is A Detector" from the Type tab of the Object Properties dialog box. Once the object is set to be a detector, the intensity detected will be displayed on the Shaded Model display, and on the text listing of the Detector Viewer window. The color scheme the detector will display on the Shaded Model may also be selected on the Object Properties dialog box. The colors chosen may represent total flux or may be normalized to be flux per unit area.

Objects acting as detectors can only display incoherent intensity. Each individual triangle used to draw the object becomes a single pixel. All rays which strike the pixel will be summed to yield the total intensity. The object detectors are cleared whenever all detectors are cleared from the detector control dialog box.

A tool for creating POB files for use as detectors is described in "Create Polygon Object".

<u>Comments on general objects as detectors pixel numbering</u>

The pixels are numbered in the order the polygons are defined, and there is no general way to describe the pixel ordering. For these objects, some experimentation may be required to determine the exact pixel ordering method.

<u>Use with LightningTrace</u>

This detector type is not supported with LightningTrace.

Tolerancing

Zemax provides a flexible and powerful tolerance development and sensitivity analysis capability. The tolerances available for analysis include variations in construction parameters such as curvature, thickness, position, index of refraction, Abbe number, aspheric constants, and much more. Zemax also supports analysis of decentration of surfaces and lens groups, tilts of surfaces or lens groups about any arbitrary point, irregularity of surface shape, and variations in the values of any of the parameter or extra data. Since the parameter and extra data terms may describe aspheric coefficients, gradient index coefficients, and more, any of these values may also be made part of the tolerance analysis. The various tolerances may be used in any combination to estimate alignment and fabrication error effects on system performance.

> **Zemax always uses exact ray tracing for tolerance analysis; there are no approximations or extrapolations of first order results in the Zemax tolerance algorithms.**

Tolerances are defined using operands, such as TRAD, which defines a tolerance on a radius. The tolerance operands are automatically saved with the lens file. Tolerance operands are edited on the Tolerance Data Editor, available from the Editors group of the "Setup Tab" or in the "Tolerance Tab".

Tolerances may be evaluated by several different criterion, including RMS spot radius, RMS wavefront error, MTF response, boresight error, user defined merit function, or a script which defines a complex alignment and evaluation procedure. Additionally, compensators may be defined to model allowable adjustments made to the lens after fabrication. Zemax also allows limits to be placed on the change of a compensator.

Tolerances may be computed and analyzed three ways:

Sensitivity Analysis: For a given set of tolerances, the change in the criterion is determined for each tolerance individually. Optionally, the criterion for each field and configuration individually may be computed.

Inverse Sensitivity: For a given permissible change in criterion, the limit for each tolerance is individually computed. Inverse sensitivity may be computed by placing a limit on the change in the criterion from nominal, or by a limit on the criterion directly. The criterion may be computed as an average over all fields and configurations, or on each field in each configuration individually.

Monte Carlo Analysis: The sensitivity and inverse sensitivity analysis considers the effects on system performance for each tolerance individually. The aggregate performance is estimated by a root-sum-square calculation. As an alternative way of estimating aggregate effects of all tolerances, a Monte Carlo simulation is provided. This simulation generates a series of random lenses which meets the specified tolerances, then evaluates the criterion. No approximations are made other than the range and magnitude of defects considered. By considering all applicable tolerances simultaneously and exactly, highly accurate simulation of expected performance is possible. The Monte Carlo simulation can generate any number of designs, using normal, uniform, parabolic, or user defined statistics.

The basic procedure

Tolerancing a lens consists of these steps:

1) Define an appropriate set of tolerances for the lens. Usually, the default tolerance generation feature described in the "Tolerance Wizard" section of the "Tolerance Tab" help file is a good place to start. Tolerances are defined and modified on the Tolerance Data Editor, available in the Editors group of the "Setup Tab" or in the "Tolerance Tab".

2) Modify the default tolerances or add new ones to suit the system requirements.

3) Add compensators and set allowable ranges for the compensators. The default compensator is the back focal distance, which controls the position of the image surface. Other compensators, such as image surface tilt, may be defined. There is no limit to the number of compensators that may be defined.

4) Select an appropriate criterion, such as RMS spot radius, wavefront error, MTF, or boresight error. More complex criterion may be defined using a user defined merit function, or for comprehensive flexibility, a tolerance script, which is described later.

5) Select the desired mode, either sensitivity or inverse sensitivity. For inverse sensitivity, choose criterion limits or increments, and whether to use averages or compute each field individually.

6) Perform an analysis of the tolerances.

7) Review the data generated by the tolerance analysis, and consider the budgeting of tolerances. If required, modify the tolerances and repeat the analysis.

Details on this basic procedure will be provided in subsequent sections.

Tolerance operands

A tolerance operand has a four letter mnemonic, such as TRAD for Tolerance Radius. Three integer values, abbreviated Int1, Int2, and Int3, are associated with the mnemonic to identify the surface or surfaces of the lens to which the tolerance applies. Some tolerance operands use the Int numbers for purposes other than defining surface numbers as indicated in the following table.

Each tolerance operand also has a minimum and maximum value. These values refer to the maximum acceptable change from the nominal value. Each operand also has space for an optional comment to make the tolerance set easier to read. The available tolerance operands are listed in the following table, and are described in detail below.

Tolerance Operands Summary Table

Name	Int1	Int2	Int3	Description
Surface Tolerances				
TRAD	Surf #	-	-	Tolerance on radius in lens units.
TCUR	Surf #	-	-	Tolerance on curvature in inverse lens units.
TFRN	Surf #	-	-	Tolerance on radius in fringes.
TTHI	Surf #	Adjust #	-	Tolerance on thickness or position in lens units, with optional adjuster.
TCON	Surf #	-	-	Tolerance on conic constant t (dimensionless).
TSDI	Surf #	-	-	Tolerance on semi-diameter in lens units.
TSDX	Surf #	-	-	Tolerance on Standard surface x-decenter in lens units.
TSDY	Surf #	-	-	Tolerance on Standard surface y-decenter in lens units.
TSTX	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in x in degrees.
TSTY	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in y in degrees.
TIRX	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in x in lens units.
TIRY	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in y in lens units.
TIRR	Surf #	-	-	Tolerance on stantard surface irregularity.
TEXI	Surf #	Max Term#	Min Term#	Tolerance on surface irregularity using Fringe Zernikes.
TEZI	Surf #	Max Term#	Min Term#	Tolerance on surface irregularity using Standard Zernikes.
TPAI	Surf #	Parameter #	-	Tolerance on the inverse of parameter value of surface.
TPAR	Surf #	Parameter #	-	Tolerance on parameter value of surface.

TEDV	Surf #	Extra Data #	-	Tolerance on extra data value of surface.
TIND	Surf #	-	-	Tolerance on index of refraction at d light
TABB	Surf #	-	-	Tolerance on Abbe Vd number
TCMU	Surf #	Layer #	-	Tolerance on coating multiplier.
TCIO	Surf #	Layer #	-	Tolerance on coating index offset.
TCEO	Surf #	Layer #	-	Tolerance on coating extinction offset.
Element Tolerances				
TEDX	First Surf	Last Surf	-	Tolerance on element x- decenter in lens units.
TEDY	First Surf	Last Surf	-	Tolerance on element y- decenter in lens units.
TETX	First Surf	Last Surf	-	Tolerance on element tilt about x in degrees.
TETY	First Surf	Last Surf	-	Tolerance on element tilt about y in degrees.
TETZ	First Surf	Last Surf	-	Tolerance on element tilt about z in degrees.
User Defined Tolerances				
TUDX	Surf #	-	-	Tolerance on user defined decenter x.
TUDY	Surf #	-	-	Tolerance on user defined decenter y.
TUTX	Surf #	-	-	Tolerance on user defined tilt x.
TUTY	Surf #	-	-	Tolerance on user defined tilt y.
TUTZ	Surf #	-	-	Tolerance on user defined tilt z.
Non-Sequential Component Tolerances				
TNMA	Surf #	Object #	Data	Tolerance on NSC material Nd and Vd. The data code is 1 for Nd, and 2 for Vd.
TNPS	Surf #	Object #	Data	Tolerance on NSC object position. The data code is 1-6 for x, y, z, tilt x, tilt y, tilt z, respectively.
TNPA	Surf #	Object #	Parameter	Tolerance on NSC object parameter.

For each tolerance, a minimum and a maximum value are specified on the Tolerance Data Editor. The tolerances are described in detail below.

TRAD: Tolerance on radius

Used to tolerance directly on the radius of curvature. Min and max are the extreme errors in lens units.
For example, if the nominal radius of a surface is 100 mm, and the min and max TRAD values for that surface are -.50 mm and +.50 mm, then the tolerance analysis will be performed with the radius of the surface set to 99.50 mm and 100.50 mm.

TCUR: Tolerance on curvature

Used to tolerance in units of curvature, which is directly related to power. Min and max are the extreme errors in inverse lens units.
For example, if the nominal radius of a surface is 100 mm, then the nominal curvature is 0.01 inverse mm. If the min and max TCUR values for that surface are -.001 and +.001 inverse mm, then the tolerance analysis will be performed with the radius of the surface set to 111.11 mm and 90.909 mm.

TFRN: Tolerance on fringes

Fringe tolerances are very useful when tolerancing flat or large radii surfaces. Min and max are the extreme errors in (dimensionless) fringes. The TWAV operand is used to define the test wavelength. The change in the sag of a surface for small changes in curvature is given approximately by

$$\Delta Z = \frac{r^2}{2} \Delta C,$$

where r is the semi-diameter of the surface. The change in the sag is related to the error in fringes by

$$\Delta Z = \frac{\lambda}{2} N,$$

where N is the number of fringes. The factor of one half assumes a double pass Newton's rings type test. For more information see Malacara, Optical Shop Testing.

TTHI: Tolerance on thickness

TTHI is used to tolerance both absolute positions of elements as well as thicknesses of lenses within element groups.

When Zemax creates the default tolerances, it is assumed that all variations in thickness affect only that surface and any surfaces in contact with that element. For example, if the first lens in a contact doublet has a +1.0 mm change in thickness, the front and rear vertex of the second lens both shift by +1.0 mm.

However, since Zemax defines the position of all surfaces by using an offset from the previous surface, adding 1.0 mm to the surface will shift all subsequent lenses in the system by +1.0 mm. What is more likely to occur in fabrication is that the +1.0 mm offset would be absorbed by the first air space after the lens group. TTHI can handle this case by allowing a "adjustment" surface to be specified. When Zemax creates the default tolerances, the adjustment surface is specified as the first air space which follows the surface being toleranced.

To illustrate, imagine a lens where surface 3 was made of BK7, and surface 4 was made of F2, and surface 5 was air. The nominal thicknesses are 3, 4, and 6 mm, respectively. If a TTHI operand was defined by the default tolerance algorithm for surface 3, an adjustment would be defined for surface 5. If the tolerance value was +.1 mm, then during analysis the thicknesses would be changed to 3.1, 4.0, and 5.9, respectively. Thus, the absolute positions of surfaces 6 through the image surface are unaffected by the change in thickness on surface 3.

The adjustment is optional; to disable it, set the adjustment to the same surface number as the tolerance, such as TTHI 3 3. For some lens systems, such as those that are assembled by stacking spacers in a tube, the adjustment may not be desired.

Int1 is used to define the surface number, Int2 in the adjustment surface number, unless Int2 is equal to Int1. Min and max are the extreme errors in lens units.

TCON: Tolerance on conic

TCON is used to define a tolerance on a conic constant. Min and max are the extreme errors, dimensionless.

TSDI: Tolerance on semi-diameter

TSDI is used to define a tolerance on the semi-diameter of a surface. Min and max are the extreme errors in lens units.

TSDX, TSDY: Tolerance on surface decenters

TSDX and TSDY are used to analyze decentration in X and Y respectively of a Standard surface type. The Int1 value indicates the number of the surface, and this surface must be a Standard surface type. Surfaces other than Standard surface types may be toleranced using the TEDX and TEDY operands described later. The min and max values are the decentering in lens units.

The analysis of TSDX and TSDY uses the Irregular surface type. For more information, search the help files for the discussion "Tolerancing with the Irregular surface type".

TSTX, TSTY: Tolerance on surface tilts

TSTX and TSTY are used to analyze tilts of a Standard surface about the X and Y axes, respectively. The Int1 value indicates the number of the surface, and this surface must be a Standard surface type. Surfaces other than Standard surface types may be toleranced using the TETX and TETY operands described later.

The min and max values are the tilt in degrees about the local X and Y axis of the lens. For more information, search the help files for the description of the related TIR operands TIRX and TIRY.

The analysis of TSTX and TSTY uses the Irregular surface type. For more information, search the help files for the discussion "Tolerancing with the Irregular surface type".

TIRX, TIRY: Tolerance on surface TIR

TIRX and TIRY are used to analyze tilts of a Standard surface along the X and Y axes, respectively. The Int1 value indicates the number of the surface, and this surface must be a Standard surface type. Surfaces other than Standard surface types may be toleranced using the TETX and TETY operands described later.

TIRX and TIRY are used to specify a tolerance on total indicator runout, or TIR, which measures the amount of "wedge" in a lens.

The min and max values are twice the amount of "sag" in lens units measured at the maximum radial aperture of the surface where the maximum radial aperture is defined by the semi-diameter of the surface. The change in sag as a function of the x or y normalized coordinate for a TIRX or TIRY value is given by:

$$\Delta Z_x = \frac{TIRX}{2}\rho_x; \ \Delta Z_y = \frac{TIRY}{2}\rho_y.$$

For example, if the TIRX tolerance is 0.10 mm, then the change in sag at the maximum +x aperture of the lens is 0.05 mm, and the deviation at the maximum -x aperture is -.05 mm, for a "total" TIR of 0.10 mm. A similar discussion applies to TIRY. The min and max values are used to model the tilt of the surface in each direction. The tilt angle that is actually placed on the surface is given by

$$\theta_x = \tan^{-1}\left(\frac{\Delta Z_y}{S}\right); \ \theta_y = \tan^{-1}\left(\frac{\Delta Z_x}{S}\right),$$

where S is the semi-diameter of the surface. Note that a sag along Y implies a rotation about the X axis, and a sag along X implies a rotation about the Y axis.

The analysis of TIRX and TIRY uses the Irregular surface type. For more information, search the help files for the discussion "Tolerancing with the Irregular surface type".

TIRR: Tolerance on surface irregularity

> **For a more detailed treatment of irregularity, search the help files for the discussion on the TEXI and TEZI operands which follows.**

TIRR is used to analyze irregularity of a Standard surface. The Int1 value indicates the number of the surface, and this surface must be a Standard surface type. Analysis of irregularity on surface types other than Standard is not directly supported.

Modeling irregularity is somewhat more problematic than other types of tolerances. This is primarily because irregularity by nature is random, and not deterministic such as a change in radius. Therefore, some assumptions about the nature of the irregularity need to be made in order to perform the analysis. The assumption Zemax makes when using TIRR is that the irregularity is half spherical aberration, and half astigmatism. This is less restrictive model than assuming 100% astigmatism, because astigmatism cannot be compensated by focus, and is therefore a more serious defect in the lens.

The min and max values are the irregularity in units of fringes measured at the maximum radial aperture of the surface where the maximum radial aperture is defined by the semi-diameter of the surface. The TWAV operand is used to define the test wavelength.

Zemax assumes fringes are measured in a double pass Newton's rings type test. For example, a TIRR of "W" fringes would yield a change in sag of the surface of

$$\Delta z = \frac{\lambda_t W}{4}(\rho^4 + \rho'^2_y),$$

where λ_t is the test wavelength (defined by the TWAV operand), ρ is the normalized radial coordinate, and ρ'_y is the normalized and rotated radial coordinate as defined in "Irregular". The change in wavefront optical path is related to the change in sag and the index of refraction of the two media the surface separates:

$$\Delta OP = \Delta z(n_2 - n_1).$$

The analysis of TIRR uses the Irregular surface type.

When computing the Monte Carlo analysis, the angle of the astigmatism is chosen randomly between 0 and 360 degrees. This allows simulation of randomly oriented astigmatic error, which is less severe and more realistic than placing all the astigmatism along the y axis of each element.

Search the help files for the discussion "Tolerancing with the Irregular surface type" for more information.

TEXI: Tolerance on surface irregularity using the Fringe Zernike model

See also the discussion for TEZI. TEZI is a superior alternative to TEXI.

TEXI is used to analyze random irregular deviations of small amplitude on a surface that is either a Standard, Even Aspheric, or Zernike Fringe Sag surface. Analysis of irregularity on surface types other than Standard, Even Aspheric, or Zernike Fringe Sag is not directly supported. The Int1 value indicates the number of the surface, Int2 defines the maximum Fringe Zernike term (must be between 3 and 37), and Int3 defines the minimum Fringe Zernike term (must be between 2 and the maximum term).

TEXI uses the Zernike Fringe Sag surface (search the help files for "Zernike Fringe Sag") to model the irregularity rather than using the third order aberration formulas used by TIRR. When using TEXI, the min and max tolerance values are interpreted to be the approximate magnitude of the zero to peak error of the surface in double-pass fringes at the test wavelength. The zero to peak is only a very rough measure of the irregularity. Whether the zero to peak and peak to valley are the same depends upon the particular Zernike term used. The min tolerance value is automatically set to the negative of the max value; this is done to yield both positive and negative coefficients on the Zernike Fringe Sag surface. The vagueness of defining irregularity by the zero to peak measure is the prime disadvantage to TEXI, and the reason why the newer TEZI operand is the preferred alternative.

Zemax computes the coefficients on the individual terms of the Zernike polynomial that defines the deviation in the shape of the surface using the following formula:

$$c = \frac{\lambda}{2} \frac{f}{\sqrt{n}},$$

where f is the number of double pass fringes, n is the number of Zernike terms used (note n is given by Int2 - Int3 + 1), and λ is the test wavelength. The coefficients are scaled by one over root n to account for the fact that a random collection of Zernike terms will generally sum in an RSS sense; so the PTV error is not linear with the number of terms. Since it is convenient to specify the approximate overall PTV, the terms on each Zernike are computed according to the formula above. Note there is a "c" value for both the min and the max tolerance in fringes.

For the sensitivity analysis, the surface is converted to a Zernike Fringe Sag surface and all the coefficients of the Zernike polynomial are set to the "c" value from the above equation. Note that since the maximum surface sag deformation is the same for all Zernike Fringe polynomial terms at the edge of the aperture, a "c" value of 0.001 when using 20 Zernike terms will yield a maximum sag deviation of 20c.

For the Monte Carlo analysis, the surface is converted as for the sensitivity analysis, but each polynomial term is assigned a coefficient randomly chosen which lies between the min and max "c" tolerance values. The random value is chosen using the statistical model selected for the operand; search the help files for the STAT command for a discussion.

Generally speaking, if lower order terms are used, the irregularity will be of low frequency, with fewer "bumps" across the surface. If higher order terms are used, there will be higher frequency irregularity, with more "bumps" across the surface. Note the TIRR irregularity operand models the lowest frequency form of irregularity, with just a quadratic and quartic deviation across the surface. TEXI can model much more irregular surfaces, and with 30 or more terms used, about 5-15 "bumps" will typically be seen over the surface.

Because the Zernike Fringe Sag surface sag expression contains portions of both the Standard and Even Aspheric surfaces, either of these surface types may be modeled by the Zernike Fringe Sag surface created with the TEXI operand. If the surface already is a Zernike Fringe Sag surface, then the deviations are added to the polynomial terms already there. If the surface is either a Standard or Even Aspheric surface, the normalization radius of the Zernike Fringe Sag surface is set to the semi-diameter of the surface. If the surface was already a Zernike Fringe Sag surface, then the min and max tolerances are assumed to be measured at the normalization radius already defined.
TEXI always ignores Zernike term 1, the piston term, and sets this value to zero.

TEZI: Tolerance on surface irregularity using the Standard Zernike model

See also the discussion for TEXI. TEZI is a superior alternative to TEXI.

TEZI is used to analyze random irregular deviations of small amplitude on a surface that is either a Standard, Even Aspheric, or Toroidal surface. Analysis of irregularity on surface types other than these is not directly supported. The Int1 value indicates the number of the surface, Int2 defines the maximum Standard Zernike term (must be between 3 and 231), and Int3 defines the minimum Standard Zernike term (must be between 2 and the maximum term).

TEZI uses the Zernike Standard Sag surface (search the help files for "Zernike Standard Sag") to model the irregularity on Standard and Even Aspheric surfaces, while Toroidal surfaces use the Zernike terms already supported by the Toroidal surface. When using TEZI, the max tolerance value is the exact RMS error of the surface in lens units. The min tolerance value is automatically set to the negative of the max value; this is done to yield both positive and negative coefficients for the Zernike terms. The resulting RMS is of course always a positive number whose magnitude is equal to the max tolerance value.

For the sensitivity analysis, the surface is converted to a Zernike Standard Sag or Toroidal surface and all the coefficients of the Zernike polynomial for terms greater than #1 (the "piston" term) are set to a value so that the square root of the sum of the squares of the coefficients yields the specified RMS value. All coefficients are set to the same value.

For the Monte Carlo analysis, the surface is converted as for the sensitivity analysis, but each polynomial term is assigned a coefficient randomly chosen between -1.0 and 1.0, and the resulting coefficients are then normalized to yield the exact RMS tolerance. The random value is chosen using the statistical model selected for the operand; search the help files for the STAT command for a discussion.

The number of terms is given by Int2 - Int3 + 1. Generally speaking, if lower order terms are used, the irregularity will be of low frequency, with fewer "bumps" across the surface. If higher order terms are used, there will be higher frequency irregularity, with more "bumps" across the surface. Note the TIRR irregularity operand models the lowest

frequency form of irregularity, with just a quadratic and quartic deviation across the surface. TEZI can model much more irregular surfaces.

Because the Zernike Standard Sag surface sag expression contains portions of both the Standard and Even Aspheric surfaces, either of these surface types may be modeled by the Zernike Standard Sag surface created with the TEZI operand. If the surface is Toroidal, the Toroidal surface is retained since the Zernike terms are already supported with this surface type, however, the nominal value of all the Zernike terms must be zero if the nominal surface is Toroidal. The normalization radius for the Zernike terms is set to the semi-diameter of the surface.

TEZI always ignores Zernike term 1, the piston term, and sets this value to zero.

TPAI: Tolerance on the inverse of parameter data

TPAI is used to tolerance on the inverse of the parameter data. The Int1 number is the surface number. The Int2 number indicates the parameter number. Search the help files for "Surface Types" for tables listing which parameter numbers are used for each surface model. TPAI is very similar to TPAR, except the tolerance deviations are made on the inverse of the parameter value rather than the parameter directly. The toleranced parameter values are defined by

$$P' = \frac{1}{\left(\frac{1}{P}\right) + \Delta}$$

where P is the parameter value, P' is the perturbed parameter value, and Δ is the amount of tolerance perturbation. Min and max are the changes measured in the inverse of whatever units are used by the parameter.

TPAR: Tolerance on parameter data

TPAR is used to tolerance parameter data. The Int1 number is the surface number. The Int2 number indicates the parameter number. Search the help files for "Surface Types" for tables listing which parameter numbers are used for each surface model. TPAR is useful for determining tolerances on aspheric coefficients, and more. Min and max are the changes in whatever units are used by the parameter.

TEDV: Tolerance on extra data values

TEDV is the tolerance for extra data values. Extra data values are used by some surfaces in the lens data editor in Zemax- Professional to define aspheric terms, diffraction coefficients, or other data. Int1 identifies the surface, Int2 identifies the extra data number, and min and max are the extreme errors in whatever units are used by the extra data value. Search for the "Surface Types" reference file for tables listing which extra data numbers are used for each surface model.

TIND: Tolerance on index

TIND is the tolerance on index of refraction. Int1 identifies the surface, and min and max are the extreme errors in index of refraction. The index error is modeled as an "offset" which is independent of wavelength, unless either the glass of the surface is a "model" glass defined by the index at d-light, Abbe number, and dPgF, or the glass of the surface is a catalog glass and all defined wavelengths are between 0.3 and 2.5 micrometers. In either of these latter cases, the TIND is

interpreted to be a change in the d-light index, which will change the index at all wavelengths in a nonlinear fashion. The change in index is given by the change in the model glass index for the same change in the d-light index as a function of wavelength. For details on the model glass, search the help files for "Using model glasses".

TABB: Tolerance on Abbe

TABB is the tolerance on the Abbe or Vd number. Int1 identifies the surface, and min and max are the extreme errors in the Abbe number. If either the glass of the surface is a "model" glass defined by the index at d-light, Abbe number, and dPgF, or the glass of the surface is a catalog glass and all defined wavelengths are between 0.3 and 2.5 micrometers, TABB is interpreted to be a change in the d-light Abbe number, which will change the index at all wavelengths in a nonlinear fashion. Otherwise, TABB is ignored.

The change in index is given by an estimate of the differential change in the index at any wavelength as a function of the d-light index, Abbe, and dPgF.

TCMU: Tolerance on coating multiplier

Coating multipliers are described in "Optimizing coatings with Zemax" and "Surface coating tab". The TCMU operand allows tolerancing of the coating multipliers, which in effect places a tolerance on the thickness of individual layers in the optical coating on a surface.

TCIO: Tolerance on coating index offset

Coating index offsets are described in "Optimizing coatings with Zemax" and "Surface coating tab". The TCIO operand allows tolerancing of the coating index offsets, which in effect places a tolerance on the real part of the material index of individual layers in the optical coating on a surface.

TCEO: Tolerance on coating extinction offset

Coating multipliers are described in "Optimizing coatings with Zemax" and "Surface coating tab". The TCEO operand allows tolerancing of the coating extinction offsets, which in effect places a tolerance on the imaginary part of the material index of individual layers in the optical coating on a surface.

TEDX, TEDY: Tolerance on element decenters

TEDX and TEDY are used to analyze decentration in X and Y respectively. The two surface numbers defined by Int1 and Int2 indicate the beginning and ending surfaces of a lens group. The min and max values are the decentering in lens units.

These tolerances require Zemax to insert a coordinate break surface before and after the group, and then decenter the entire group as a unit. For this reason, TEDX and TEDY generally should not be used on either side of an existing coordinate break. If you require this capability, search the help files for the discussion on TUDX and TUDY.

TEDX and TEDY may also be used to model decenter of surfaces, like TSDX and TSDY. TEDX and TEDY will work for surfaces of any type, including Standard surfaces and non-Standard surfaces, while TSDX and TSDY only work for Standard surfaces. To decenter a single surface using TEDX or TEDY, set Int1 and Int2 to the same surface number.

To check to see if Zemax is actually doing what you want, search the help files for the SAVE command discussed in the section "Trouble shooting the tolerance results".

TEDX and TEDY operands may be nested. For example it is possible to analyze the decentration of one surface group defined by surfaces 5 through 20; while simultaneously analyzing decenters of elements defined by surfaces 5-8, 8-12, 14-20, etc. This capability can simulate the alignment errors of elements within an assembly, as well as the alignment error of the assembly as a whole. The nesting rules are fully described in the section "Nesting rules for Monte Carlo analysis".

TETX, TETY, TETZ: Tolerance on element tilts

TETX, TETY, and TETZ are used to analyze tilts of either a surface or a lens group about the X, Y, or Z axes, respectively. The two surface numbers defined by Int1 and Int2 indicate the beginning and ending surfaces of a lens group. The min and max values are the tipping angles in degrees.

These tolerances require Zemax to insert a coordinate break surface before and after the lens group, and to use a dummy surface at the end of the group to return to the front vertex. The entire group may then be pivoted about a point as a unit. For this reason, TETX/Y/Z should not be used when the range of surfaces defined includes a coordinate break which is related to a following coordinate break outside the surface range controlled by the TETX/Y/Z by pickup solves. The hazard is if the two resulting tilt ranges overlap, then the position of the elements may not be what is intended. If you require this capability, search the help files for the discussion on TUTX, TUTY, and TUTZ.

TETX and TETY may also be used to model tilt of single surfaces, sometimes called "wedge", like TSTX and TSTY. TETX and TETY will work for surfaces of any type, including Standard surfaces and non-Standard surfaces, while TSTX and TSTY only work for Standard surfaces. To tilt a single surface using TETX or TETY, set Int1 and Int2 to the same surface number.

TETX and TETY by default pivot about the front vertex of a lens group, however it is often advantageous to pivot about some other point. For example, well-designed lens mounts will pivot about the nodal point of a lens to maintain focus during alignment. This case is easily modeled by Zemax through the use of a dummy surface at the nodal point. Change the starting surf (Int1) to be at the nodal point dummy surface, and the pivoting will be about that point. The first surface may be located anywhere with respect to the rest of the lens group; and so the tilt may be about any point.

TETX, TETY, and TETZ operands may be nested. For example it is possible to analyze the tilt of one surface group defined by surfaces 5 through 20; while simultaneously analyzing tilts of elements defined by surfaces 5-8, 8-12, 14-20, etc. This capability can simulate the alignment errors of elements within an assembly, as well as the alignment error of the assembly as a whole. The nesting rules are fully described in the section "Nesting rules for Monte Carlo analysis".

TOFF: Tolerance off (can be used for comments)

This operand is a place holder that has no effect on the tolerance analysis. The TOFF operand may be used to enter comments.

TUDX, TUDY, TUTX, TUTY, TUTZ: Tolerance on user defined tilts and decenters

These five tolerances, TUDX, TUDY, TUTX, TUTY, and TUTZ, are used for more general user defined tilts and decenters. The names are mnemonics for Tolerance User Decenter/Tilt X, Y, and Z. These are very similar to the TEDX, TEDY, TETX, and TETY tolerances. The difference is that Zemax does not automatically insert the required coordinate break surfaces to achieve the specified decenters and tilts in the tolerances. To use the TUDX, TUDY, TUTX, TUTY, and TUTZ commands, you must have already defined the surface specified by Int1 to be a coordinate break surface. Normally, but not always, there will be a second coordinate break later in the Lens Data Editor that has a pickup solve on the toleranced parameter. The pickup solve may be positive or negative, as the case warrants. This permits some complex pivoting and decentering about arbitrary points, with a certain finesse required. Min and max are in lens units for TUDX and TUDY, and in degrees for TUTX, TUTY, and TUTZ, just like the coordinate break surface.

TNPS, TNPA, TNMA: Tolerances on non-sequential data

TNPS defines tolerances on Non-Sequential Component (NSC) data. Three integer parameters are required: the NSC surface number (use 1 if the program mode is NSC), the object number, and a code. The code is 1, 2, or 3 for the object's X, Y, or Z position, respectively; the code is 4, 5, or 6 for the X-Tilt, Y-Tilt, or Z-Tilt, respectively.

TNPA defines tolerances on NSC parameter data. The surface and object integers are defined in the same way as for TNPS; the third integer is for the parameter number. Search the help files for "NSC Objects" for a detailed explanation of what parameter numbers are used by which object types.

TNMA defines tolerances on the index Nd and Abbe Vd of NSC material data. Three integer parameters are required: the NSC surface number (use 1 if the program mode is NSC), the object number, and a code. The code is 1 or 2 for the material Nd and Vd values, respectively. When using TNMA, the change in the material index is computed exactly as for the sequential glass offset solve, as described in "Glass: Offset".

TMCO: Tolerance on multi-configuration data

TMCO specifies the tolerance on any numerical value in the Multi-Configuration Editor. The two integer arguments are the operand row number and configuration number.

Tolerancing with the Irregular surface type

The tolerance operands TSDX, TSDY, TSTX, TSTY, and TIRR all use the Irregular surface type to model the perturbation to the lens surface. For a detailed description of the Irregular surface type.
The overwhelming advantages of using the Irregular surface type are speed, simplicity, and flexibility. Any Standard surface type can be converted to an Irregular surface type without the need for dummy surfaces or coordinate breaks. Furthermore, the aggregate effects of tilt, decenter, and irregularity may all be modeled simultaneously using the Monte Carlo analysis.

When Zemax computes the tolerance analysis using the operands TSDX, TSDY, TSTX, TSTY, or TIRR, the surface is first converted from a Standard to an Irregular surface type. This is why only Standard surface types are supported when using these operands.

Tolerance control operands

There are also a few tolerance control operands which can be entered in the Tolerance Data Editor. These operands are not tolerances, but are used to define compensators, to save intermediate results for further evaluation, to define statistical properties, and to define the test wavelength for fringe tolerances.

Tolerance Control Operands Summary Table

Name	Int1	Int2	Description
CEDV	Surface #	Extra Data #	Sets an extra data value as a compensator.
CMCO	Operand #	Config #	Sets a multi-configuration operand value as a compensator. Search the help files for the discussion on CMCO below for important information.
CNPA	Object #	Parameter #	Sets a non-sequential parameter as a compensator. Int3 is the surface number, use 1 for non-sequential mode.
CNPS	Object #	Code	Sets a non-sequential object position value as a compensator. Code is 1-6 for x, y, z position and x, y, z tilt, respectively. Int3 is the surface number, use 1 for non- sequential mode.
COMM	-	-	This operand is used to print the comment, or if no comment is defined, to print a blank line in the tolerance analysis output report.
COMP	Surface #	Code	Sets a compensator. Code is 0 for thickness, 1 for curvature (1 over the radius), 2 for conic.
CPAR	Surface #	Parameter #	Sets a parameter as a compensator. For a list of parameters that control surface attributes, search the help files for "Surface Types".
SAVE	File Number	-	Saves the file used to evaluate the tolerance on the prior row in the editor. See the discussion below.
SEED	Seed #	-	Seeds the random number generator for Monte Carlo analysis. Use a value of 0 to choose a random seed.
STAT	Type	# of standard deviations	Sets the type of statistical distribution for randomly selected Monte Carlo parameter analysis. See the discussion below.
TWAV	-	-	This operand sets the test wavelength. The "Min" column is used to edit and display the test

			wavelength.

The tolerance control operands are described in detail below.

CEDV: Define extra data value compensator

CEDV is used to define the surface number and extra data number of the compensator to use for tolerance analysis. Int1 is used to specify the surface, and Int2 is used to define the extra data number. For example, to specify extra data value 17 of surface 9 is a compensator, use CEDV with Int1 = 9, Int2 = 17.

Search the help files for "Surface Types" for tables defining which extra data values are used by which surface types.

General comments about min and max values on compensators

The min and max values of compensator operands such as CEDV, CMCO, COMP, and CPAR indicate the maximum change allowed for the compensator. For example, if the nominal value of a compensator is 50, and the min and max values are -1.0 and 1.0, then the compensator will be restricted to lie between 49.0 and 51.0. This bounding of the compensators is ignored if "merit function" or "user script" is selected as the tolerance criterion, although the compensators will still be used.

Note that while min and max values serve to define boundaries for the compensator, these boundaries may be violated if doing so would provide better performance of the system overall (just as with boundary operands used during optimization; see "Understanding boundary operands" in the Optimization Overview reference file section). If you find that a compensator boundary is being violated consistently during your tolerance analysis, you may wish to investigate whether that boundary is reasonable for the defined compensator. Using the SAVE tolerance operand can help with such an analysis (see "SAVE" operand under the "Tolerance control operands" section).

There is no limit to the number of compensators that may be defined.

CMCO: Define multi-configuration operand compensator

CMCO is used to define the operand number and configuration number of the compensator to use for tolerance analysis. Int1 is used to specify the operand number, and Int2 is used to define the configuration number. For example, to specify multi-configuration operand 6 in configuration 4 is a compensator, use CMCO with Int1 = 6, Int2 = 4. The order of the CMCO operands matters. The CMCO operands must be defined in order of operand number then configuration. All CMCO operands for MCE operand 1 must precede CMCO operands for MCE operand 2, etc. Within each group of CMCO operands using the same MCE operand number, the CMCO operands must be listed in order of configuration number.

Search the help files for "Using Multiple Configurations" for information on using multi-configuration operands.

Also, search the help files for "General comments about min and max values on compensators".

CNPA: Define non-sequential parameter compensator

CNPA is used to define the object and parameter number of the compensator to use for tolerance analysis. Int1 is used to specify the object number, and Int2 is used to define the parameter number. Int3 is the surface number, use 1 for non-sequential mode.

Also, search the help files for "General comments about min and max values on compensators".

CNPS: Define non-sequential position compensator

CNPS is used to define the object and position code of the compensator to use for tolerance analysis. Int1 is used to specify the object number, and Int2 is used to define the position code. Int3 is the surface number, use 1 for non-sequential mode. The position code is 1-6 for x, y, z position and x, y, z tilt, respectively.

Also, search the help files for "General comments about min and max values on compensators".

COMP: Define compensator

COMP is used to define the surface number and type of compensator to use for tolerance analysis. Int1 is used to specify the surface, and Int2 is used to define the type. Int2 uses a "code" defined as follows:

Int2 = 0, compensator is thickness
Int2 = 1, compensator is curvature (1 over the radius); note that min and max values defined for the compensator should also be in units of curvature
Int2 = 2, compensator is conic

Also, search the help files for "General comments about min and max values on compensators".

CPAR: Define parameter compensator

CPAR is used to define the surface number and parameter number of the compensator to use for tolerance analysis. Int1 is used to specify the surface, and Int2 is used to define the parameter number. For example, to specify parameter 2 of surface 5 is a compensator, use CPAR with Int1 = 5, Int2 = 2.

Search the help files for "Surface Types" for tables defining which parameters are used by which surface types.

Also, search the help files for "General comments about min and max values on compensators".

SAVE: Save sensitivity analysis lenses

The operation of the tolerance feature is not always transparent. To get a closer look at what the tolerance routine is actually doing, use the SAVE operand. SAVE can be inserted after any tolerance you would like to inspect in more detail. For example, suppose you had the tolerance operand TEDX in the Tolerance Data Editor. After reviewing the resulting sensitivities, the results do not appear to make sense. Edit the tolerance operands in the Tolerance Data Editor to add the SAVE command after the TEDX command. For the Int1 value, enter in any integer between 1 and 9999, inclusive. If the integer number is 0, no file is saved. The next time the tolerance analysis is run, Zemax will save the files used to compute the TEDX tolerance. The file names will be TSAV_MIN_xxxx.ZMX and

TSAV_MAX_xxxx.ZMX for the min and max tolerance analysis, respectively, where xxxx is the integer number specified in the Int1 column.

The lens files will be saved in the same folder as the current lens. Zemax will save the lens used to evaluate the min and max values of the preceding tolerance in a ZMX format file. Run the tolerance file in the usual way, and then load one of the files generated. The file will contain the prescription data, coordinate breaks, pickups, variables (which are optimally adjusted compensators), and merit function used to determine the criterion data. This procedure will let you verify the configuration is in fact what you are trying to model, and will give you insight into how Zemax adjusts your lens prescription to determine the tolerances.

SEED: Seed the random number generator

By default, every Monte Carlo analysis will produce new, randomly generated systems. There are times when it is useful to have multiple Monte Carlo analysis runs produce the same perturbations every time. The SEED operand may be used for this purpose. If the SEED is any value other than zero, the random number generator will be seeded with the specified value. This value is automatically incremented for each generated system with a single Monte Carlo analysis so that each generated system is unique, yet random and reproducible. Using a value of 0 will select a random seed based upon the system clock, which will generally not be the same each time the Monte Carlo analysis is run.

The placement of the SEED operand matters. All tolerance operands listed under a SEED operand will be affected by the seed value provided. Multiple SEED operands may be placed in the tolerance data editor, so that different operands may use the same seed every time, while others may use random seeds.

STAT: Define statistics

STAT is used to define the statistics "on the fly" during the Monte Carlo analysis. The STAT command takes two integer arguments. Int1 specifies the statistics type: 0 for normal distribution, 1 for uniform distribution, 2 for parabolic distribution, and 3 for user defined distribution statistics. The Int2 value is used only by the normal distribution, and it defines "n", the number of standard deviations between the mean and the tolerance extremes. This value is capped at 10 standard deviations.

The statistics types are defined in detail in "Monte Carlo analysis".

TWAV: Test wavelength

This operand sets the test wavelength. When setting the default tolerances, Zemax adds a TWAV operand with a value of 0.6328 micrometers (HeNe) for the test wavelength. If no TWAV operand is defined, then Zemax defaults to the primary wavelength. More than one TWAV may be placed in the operand list; each operand defines the test wavelength for the operands that follow. Only operands whose min and max values are measured in fringes are affected by this setting. The "Min" column in the Tolerance Data Editor is used to edit and display the test wavelength.

Defining compensators

Many different types of compensators may be defined; thicknesses (most commonly used), curvature, conic constants, any parameter or extra data value, of any surface or surfaces. Multi-configuration operands may also be defined as compensators. The parameter values are useful for using tilts and decenters of particular components for compensation. The surface to be tilted must already be defined as a coordinate break (or perhaps a tilted surface), with any appropriate pickup solves as required.

The image surface focus is specified as a compensator by default. You may add or delete compensators to customize the analysis to your particular situation, and may use as many compensators as you like. Generally, using more compensators will loosen tolerances, and complicate the actual alignment of the system.

All compensators are defined using the COMP, CPAR, CEDV, and CMCO tolerance operands. Defining compensators is described in the section "Tolerance control operands".

When using a script to define the tolerance procedure, the compensators may be altered and different sets of compensators may be defined at different stages of the simulated alignment procedure. For details on this option, search the help files for "Using Tolerance Scripts".

How Zemax computes the tolerance analysis

Zemax begins the tolerance analysis by saving the lens in a temporary file which will be used to restore the lens after the tolerancing is complete. All of the changes made during tolerancing will ultimately be discarded, and the original lens file will be restored unmodified. The exception is during inverse sensitivity analysis, where the tolerance data min and max limits may be altered.

Zemax then removes all of the variables. Solves are left in place, however, solves may cause problems during tolerancing; search the help files for "Tolerancing with solves" for more information.

The tolerance operators are read, and the compensating parameters defined by COMP and CPAR specified are set as variables. The compensators defined may be altered during the analysis if a tolerance script is being used. If ray aiming is on in the lens being toleranced, or if the "Force Ray Aiming On" switch is checked, then ray aiming will be used in evaluating the tolerances, otherwise, ray aiming is left off. Tolerances computed using ray aiming are more accurate, but execution speed is slower. Search the help files for "Ray Aiming" for details on ray aiming.

Zemax then uses the Criterion, Field, MTF Frequency, and Sampling settings on the tolerance dialog box to define an appropriate merit function for the tolerancing. Since this is done to the temporary file only, the original merit function defined for the lens is not disturbed.

Boundary constraints are added to the merit function to limit the compensators to the min and max boundaries specified using the COMP and CPAR commands. If using a user defined merit function, or if the "Comp" control is set to "Paraxial Focus", boundary constraints on the compensators are ignored.

If "Comp" is set to "Optimize All", Zemax then calls the optimization function to find the best values for the defined compensators. Otherwise, the back focal position is set to paraxial focus. The resulting lens file is then saved for subsequent use by the tolerancing algorithm.

The criterion of this lens is considered to be the "nominal". Note that the nominal criterion value generally will not be the same as the criterion value that is reported on the optimization or merit function editor windows, because Zemax constructs a new merit function just for use in tolerancing. Zemax then proceeds with either sensitivity, inverse, or Monte Carlo analysis as described in subsequent sections.

Evaluating compensators

Zemax has two methods for evaluating compensators selected by the "Comp" control: either "Optimize All" or "Paraxial Focus". If "Paraxial Focus" is selected, then several assumptions are made which greatly speeds up the tolerance evaluation. First, all defined compensators and compensator boundary constraints are ignored. Back focus is implemented as a compensator using a paraxial focus error solve. This means the focus is adjusted to maintain the amount of defocus in the nominal lens design, without attempting to reoptimize the back focus precisely. These assumptions greatly speed up the tolerance evaluation. If "Optimize All" is selected, then Zemax uses the slower but more accurate optimization algorithm to find the best value for all the compensators.

"Paraxial Focus" mode is fast, accurate, and should be used if back focus is the only compensator, and the system is well described by paraxial rays, such as systems with rotational symmetry. "Paraxial Focus" should not be used when the system is highly non-symmetric or there are multiple compensators. When in doubt, run both modes and compare the results. Since "Paraxial Focus" does not precisely optimize the compensators, "Paraxial Focus" results are generally more pessimistic than "Optimize All" results.

Sensitivity analysis

For the sensitivity analysis, each tolerance is evaluated independently using the following algorithm:

The temporary lens is restored.

The parameter whose tolerance is being evaluated is adjusted to the extreme minimum value. For example, if the tolerance being evaluated is TRAD, and the nominal value is 100 mm, with a minimum tolerance of -0.1 mm, the radius is set to 99.9. If the tolerance is an element tilt or decenter tolerance, dummy coordinate breaks are inserted as required to model the perturbation. For surface tilts and decenters, such as TSDX, TSDY, TSTX, TSTY, TIRX, or TIRY, the Irregular surface type is used if the surface is initially of type Standard.

The compensators are adjusted.

The resulting criterion are printed on the report.

The procedure is repeated for the maximum tolerance.

This basic algorithm is repeated for each tolerance operand.

The value of the sensitivity analysis is that the tolerances which are too loose will, in general, have greater contributions to the increase in the criterion than other tolerances. This technique allows the designer to identify surfaces which are highly sensitive to certain errors, such as tilts or decenters. Different surfaces also will, in general, have very different sensitivities to the various errors. The sensitivity analysis aids in identification of which tolerances need to be tightened, and which might be loosened. It is also valuable for finding the optimum (and minimum) number of compensators, and the required range of adjustment. There are in fact many more applications for this feature; for example, designing lens mounts to maximize compensation leverage.

> **The sensitivity analysis aids in identification of which tolerances need to be tightened, and which might be loosened.**

The amount of output can be overwhelming, especially for lenses with many elements and a corresponding large number of tolerances. Often, the sensitivity to the tolerances varies widely over all the possible tolerances. The "Show worst"

control is extremely useful for summarizing the worst offenders, because it sorts the tolerances by contribution to the criterion and then prints them in descending order. The "Hide all but worst" control turns off the bulk of the printing if only the worst offenders are of interest.

The RSS estimated change

After all the individual tolerances are computed, Zemax then computes a variety of statistics, the most important of which is the estimated change in the criterion and associated estimated performance. Zemax uses a Root Sum Square (RSS) assumption for computing the estimated changes in the performance. For each tolerance, the change in performance from the nominal is squared and then averaged between the min and max tolerance values. Note that the change for each individual tolerance itself can be computed as a linear difference or as an RSS difference; search the help files for "Change" under "Performing the tolerance analysis". The total change can then be computed from a sum over all tolerances of the averaged squared values as follows:

$$C^2 = \sum_i \frac{\Delta_{i,\,min}^{\,2} + \Delta_{i,\,max}^{\,2}}{2}.$$

The average of the min and max squared change is taken because the min and the max tolerance cannot both occur simultaneously, and so summing the squares would result in an overly pessimistic prediction. The final predicted, as-built performance is then computed. If the Change is computed using Linear Difference, the final criterion is defined as:

$$F = N + C,$$

where N is the nominal criterion. If the Change is computed using RSS Difference, the final performance is:

$$F = \sqrt{N^2 + C^2}.$$

Inverse sensitivity analysis

If an inverse sensitivity is being performed, the tolerances are computed the same way as for the sensitivity analysis. However, the calculation is performed iteratively inside a loop while adjustments are made to the min and max tolerances. For Inverse Limit mode, the adjustments are made until the resulting criterion is approximately equal to the Max Criterion. For Inverse Increment mode, the adjustments are made until the resulting change in criterion is approximately equal to the Increment value.

For example, if the mode is Inverse Limit, the Criterion is RMS spot radius, the nominal criterion is 0.035, and the Max Criterion is 0.050, Zemax will adjust the tolerances until the criterion is 0.050. If the mode is Inverse Increment, the Criterion is RMS spot radius, the nominal criterion is 0.035, and the Increment is 0.010, Zemax will adjust the tolerances until the merit is 0.045. For Inverse Limit, the Max Criterion must be greater than the nominal, except for MTF Criterion, in which case the Max Criterion must be less than the nominal, or an error message will be generated and the analysis terminated. For Inverse Increment, the Increment must always be positive.

For the inverse sensitivity iteration to converge to the desired tolerance, the criterion must smoothly and monotonically change with the tolerance value. For criterion such as RMS spot radius, this is normally the case. However, some criterion, most notably the MTF related tolerance criterion, may have multiple peaks and valleys as the tolerance value is changed. This is usually caused by the starting tolerance being too loose. If the inverse sensitivity cannot iterate to the desired criterion, reduce the starting tolerance limit and repeat the analysis.

If the option to "Separate Fields/Configs" is not selected, the criterion Zemax uses to perform the inverse analysis is the overall criterion, which is typically an average over all fields and configurations. The problem with using the average performance is that some fields or configurations may be significantly degraded by tolerance defects while other fields or configurations are not, and the average may not reveal the severity of the loss of performance at a few fields or configurations. If "Separate Fields/Configs" is checked, then Zemax computes the criterion at each field in each configuration individually, and verifies that each field meets either the Max Criterion or Increment value. For Inverse Increment mode, Zemax computes the nominal performance at every field position and reduces the tolerances until the criterion at every field is degraded no more than the increment value.

If the starting value of the tolerance yields performance better than the Max Criterion or Increment value would dictate, the tolerance is not adjusted. This means the tolerance will never be loosened; it can only be tightened during inverse sensitivity analysis. For example, if the nominal is 0.035, and the Max Criterion is 0.050, and the initial tolerance yields a criterion of 0.040, the tolerance will not increase. To compute the actual limit, the tolerance must first be loosened on the Tolerance Data Editor, and then inverse sensitivity repeated. This is done to prevent looser than necessary tolerances. Generally, tolerances that are looser than some reasonable value do not decrease manufacturing costs.

The estimated change in performance is computed in the same way as for the sensitivity analysis, using the newly adjusted tolerances. The inverse sensitivity analysis aids in tightening individual tolerances so no one defect contributes too much to performance degradation.

Note that the inverse sensitivity is computed for each tolerance individually. The overall degradation in performance estimate will still be given by the RSS of all the individual increases. Inverse tolerancing can be disabled for individual operands; which prevents the inverse tolerance algorithm from tightening tolerances even if the estimated performance exceeds the specified limits. The option to disable inverse tolerancing is on the "Operand Type" dialog box listed under the "Edit" menu of the Tolerance Data Editor.

Monte Carlo analysis

Unlike the sensitivity and inverse sensitivity analysis, the Monte Carlo analysis simulates the effect of all perturbations simultaneously.

The Monte Carlo analysis simulates the effect of all perturbations simultaneously.

For each Monte Carlo cycle, all of the parameters which have specified tolerances are randomly set using the defined range of the parameter and a statistical model of the distribution of that parameter over the specified range. By default, all parameters are assumed to follow the same normal distribution with a total width of four standard deviations between the extreme minimum and maximum allowed values. For example, a radius of 100.00 mm with a tolerance of +4.0/ - 0.0 mm will be assigned a random radius between 100.00 and 104.00 mm, with a normal distribution centered on 102.00 mm and a standard deviation of 1.0 mm.

This default model may be changed using the STAT command. Each tolerance operand may have a separate definition for the statistics, or operands with the same statistical distribution form may be grouped together. All tolerance operands which follow a STAT command use the statistical distribution defined by that STAT command. As many STAT commands as desired may be placed in the tolerance data editor.

The STAT command accepts two arguments, Int1 and Int2. Int1 should be set equal to 0 for normal, 1 for uniform, 2 for parabolic, and 3 for user defined statistics. For normal statistics only, the Int2 value should be set to the number of standard deviations between the mean and extreme values of the parameter.

The available statistical distributions are described below.

Normal statistical distribution

The default distribution is a modified Gaussian "normal" distribution of the form

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-x^2}{2\sigma^2}\right), \; -n\sigma \leq x \leq n\sigma .$$

The modification is that the randomly selected value x (measured as an offset from the midpoint between the two extreme tolerances) is restricted to lie within "n" standard deviations of zero. The default value of "n" is two, however "n" may be changed using the Int2 argument of the STAT command defined earlier. This is done to ensure that no selected value will exceed the specified tolerances. The standard deviation is set to half the range of the tolerances divided by n. For example, if "n" is 2, and a thickness is nominally 100 mm with a tolerance of plus 3 or minus 1 mm, then the selected value will be drawn from a normal distribution with a 101 mm mean and a range of plus or minus 2 mm, with a standard deviation of 1.0. If "n" is 5, then the standard deviation will be 0.4. The larger "n" is, the more likely the selected value will be closer to the mean value of the tolerance extremes. The smaller "n" is, the more the normal looks like the uniform distribution.

Uniform statistical distribution

The uniform distribution is of the form

$$p(x) = \frac{1}{2\Delta}, \; -\Delta \leq x \leq \Delta .$$

The Δ value is one half of the difference between the max and min tolerance values. Note that the randomly selected value will lie somewhere between the specified extreme tolerances with uniform probability.

Parabolic statistical distribution

The parabolic distribution is of the form

$$p(x) = \left(\frac{3x^2}{2\Delta^3}\right), \; (-\Delta \leq x \leq \Delta) ,$$

where Δ is defined exactly the same as for the uniform distribution. The parabolic distribution yields selected values that are more likely to be at the extreme ends of the tolerance range, rather than near the middle, as for the normal distribution.

User defined statistical distribution

The user defined statistical distribution is defined by a text file with tabulated distribution data. A general probability function may be defined as

$$p(x_i) = T_i, \; 0.0 \leq x_i \leq 1.0 ,$$

where the T values are tabulated for some number of discrete X values. This general distribution may be numerically integrated, and from the integral of the tabulated values an estimated X may be randomly generated with the statistics matching that of the tabulated distribution.

The format of the file is two columns of data as follows:

```
 X1 T1
 X1 T2
 X3 T3
 ...etc.
```

where the X values are monotonically increasing floating point numbers between 0.0 and 1.0, inclusively, and the T values are the probabilities of getting that X value. Note Zemax uses a probability distribution that covers the range from 0.0 to 1.0, and so the first X1 value defined MUST be equal to 0.0 (it may have any probability T1, including zero), and the last value defined MUST have an Xn value of 1.0. Up to 200 points may be used to define the distribution between X = 0.0 and X = 1.0; a warning will be issued if too many points are listed.

For each tolerance operand subsequently defined (until another STAT command is reached) the defined min and max values will determine the actual range of the random variable X. For example, if a value of 100.0 has a tolerance of -0.0 and +2.0, the probability distribution will extend over the range of 100.0 to 102.0.

Once the data is defined in a file, the file extension must be *.UDD and the file must be placed in the <data>\Tolerance folder (for details, search the help files for "Folders"), and the file name (with extension) must be placed in the comments column of the tolerance data editor on the same line as the STAT command. The STAT type must be set to "3".

> **The file name of the user defined probability distribution must be placed in the comments column of the tolerance data editor for the STAT operand.**

One possible distribution could be:

```
 0.0 0.0
 0.1 0.5
 0.2 1.0
 0.3 0.5
 0.4 0.0
 0.5 0.5
 0.8 4.0
 1.0 5.0
```

Note the X data values do not need to be evenly spaced; finer spacing may be used in regions of rapid change in probability. This distribution has two peaks, and the higher peak is highly skewed to the maximum side of the distribution.

The user defined statistical distribution is very flexible, and may be used to model any probability distribution, including skewed, multiply peaked, or measured statistical probability data. Multiple distributions may be defined and used in the same tolerance analysis.

Discussion of Monte Carlo analysis method

Note that going from normal to uniform to parabolic statistics yields successively a more pessimistic analysis, and thus more conservative tolerances.

For each cycle, the compensators are adjusted and then the criterion and compensator values are printed. After all of the Monte Carlo trials a statistical summary is provided.

The value of the Monte Carlo analysis is estimating the performance of the lens considering all of the tolerances simultaneously. Unlike the sensitivity analysis, which identifies the "worst offenders" in the system, the Monte Carlo analysis estimates the real-world performance of a system which meets the tolerances specified. The statistical summary provided may be highly useful for lens systems which are mass-produced. Lenses which are one of a kind, of course do not follow these statistics because of the inadequate sampling. However, the Monte Carlo analysis is still useful because it indicates the probability that a single lens will meet the required specification.

Nesting rules for Monte Carlo analysis

When performing the Monte Carlo analysis, all tolerances are considered simultaneously. It is possible to define element tilt and decenter operands that conflict or are ambiguous if certain rules are not followed.

Tolerances such as TEDX, TEDY, TETX, and TETY require Zemax to insert coordinate breaks before and after the surface group, and then tilt or decenter the group as a whole. The tilts and decenters performed by the first coordinate break must be "undone" by the second coordinate break. This can only be accomplished if the vertices of the first and second coordinate breaks are at the same location in 3D space. Zemax ensures this condition through the use of pickup and position solves.

This method will fail if the surface ranges specified by multiple tolerance operands overlap. For example, if there is a TETX on surfaces 3-8, and then a subsequent TETX on surfaces 5-12, the first coordinate break which tilts the 5-12 group will change the location of the surfaces 5-8, and the second coordinate break of the first group will be moved. In this case, Zemax cannot ensure the coordinate breaks will work as intended. In fact, overlapping coordinate breaks do not have a unique interpretation, and it is difficult to imagine an optical system where they even have a meaningful physical interpretation.

Tolerances may be nested however, because nesting implies an unambiguous order to the tilts and decenters. For example, a TETX on 5-12 followed by a TETX on 5-9 and a TETX on 10-12 is perfectly reasonable. This order would simulate the tilting of an assembly, composed of multiple elements, each of which may be tilted within the assembly itself.

The nesting rules are:

1) All element tilt and decenter tolerances must be nested.

2) The outermost pair of surfaces in each nested group must be first. Here is an example of a valid set of operands:

```
 TETX 5 12
 TETX 5 10
 TETX 11 12
```

Here is an invalid set of operands:

```
 TETX 5 12
 TETX 9 15
 TETX 5 15
```

The second operand is invalid because it partially overlaps the first (this violates rule 1). The third operand is invalid because although nested with operand 1, it is the outermost range of the two (this violates rule 2). The second operand must be deleted or modified, but the third operand could be placed in front of the first operand to make a legal operand list:

```
TETX 5 15
TETX 5 12
```

Note that an operand is considered nested even if it shares one or both surface limits with a previous operand, so that TETX 5-15 may be followed by another TETX 5-15 or by TETX 5-12 or TETX 13-15, but not TETX 4-13.

Using Tolerance Scripts

This feature is only available in the Professional and Premium editions of Zemax.

Tolerance Script overview

Tolerance scripts are macro-like command files that define a procedure to follow for evaluating the perfomance of a lens during tolerancing. Scripts allow simulation of a complex alignment and evaluation procedure for a lens. With scripts, the following actions can be taken to evaluate a perturbed lens:

-Add or remove compensators.

-Load new merit functions.

-Optimize defined compensators using any merit function.

-Monitor and report any value Zemax can compute via a merit function (which includes essentially any value Zemax can compute since the merit function may call ZPL macros). Other data, such as Zernike coefficients, may also be computed.

-Write data out to text or binary files for later analysis.

-Save lenses at any stage of the analysis to a ZMX file.

Any number of these operations may be combined in the script; so multiple merit functions and compensator groups may be defined. The script is executed many times during the evaluation of the tolerance analysis, including:

-Once to compute the nominal data.

-Twice for each tolerance operand in the sensitivity analysis, (once for the min and once for the max tolerance).

-Multiple times for each tolerance operand in the inverse sensitivity analysis (inverse sensitivity may require iteration).

-Once for each generated Monte Carlo lens.

The tolerance script files must be text files ending in the extension TSC, and be placed in the <data>\Tolerance folder (for details, search the help files for "Folders").

When using tolerance scripts, the min and max boundary values of compensators are not enforced; search the help files for "General comments about min and max values on compensators" for more information. If any compensators are defined in the script, then no compensators are allowed in the Tolerance Data Editor.

The Tolerance Script commands

The tolerance script commands are defined and described below.

!

Purpose:
Adds a comment to the script.

Syntax:
```
! A comment line!
```

Discussion:
The "!" symbol is used to define comments in the script that are ignored during execution of the script.

CEDV

Purpose:
Defines a new extra data value compensator.

Syntax:
```
CEDV surf param
```

Discussion:
CEDV defines a new extra data value compensator. The value "surf" is the surface number in the original lens file. Zemax automatically renumbers the surface if additional coordinate break or other dummy surfaces were inserted by the tolerance program. The value for "param" corresponds to the extra data number for the surface.

CLEARCOMP

Purpose:
Removes all current compensators.

Syntax:
```
CLEARCOMP
```

Discussion:

CLEARCOMP removes all current compensators. New compensators must be defined before calling OPTI-MIZE or OPTIMIZE-OD.

CLOSEFILE

Purpose:
Closes the current output file.

Syntax:
```
CLOSEFILE
```

Discussion:

CLOSEFILE closes the current output file, and no more data will be written to the output file until the next OPENFILE is executed. See OPENFILE below.

CMCO

Purpose:
Defines a new multi-configuration operand compensator.

Syntax:
```
CMCO operand config
```

Discussion:
CMCO defines a new multi-configuration operand compensator. The value "operand" is the operand (row) number in the original lens file. The value for "config" corresponds to the configuration number.

CNPA

Purpose:
Defines a new non-sequential parameter compensator.

Syntax:
```
CNPA object parameter surface
```

Discussion:
CNPA defines a new non-sequential parameter compensator. The value "object" is the object number in the original lens file. The value for "parameter" corresponds to the parameter number. Surface is the surface number of the NSC group, use 1 for non-sequential program mode.

CNPS

Purpose:
Defines a new non-sequential position compensator.

Syntax:
```
CNPS object code surface
```

Discussion:
CNPS defines a new non-sequential position compensator. The value "object" is the object number in the original lens file. The value for "code" is 1-6 for x, y, z position and x, y, z, tilt, respectively. Surface is the surface number of the NSC group, use 1 for non-sequential program mode.

COMP

Purpose:
Defines a new compensator.

Syntax:
```
COMP surf code
```

Discussion:

COMP defines a new compensator. The value "surf" is the surface number in the original lens file. Zemax automatically renumbers the surface if additional coordinate break or other dummy surfaces were inserted by the tolerance program. The value for "code" is 0 for thickness, 1 for curvature, 2 for conic.

CPAR

Purpose:
Defines a new parameter compensator.

Syntax:
```
CPAR surf param
```

Discussion:
CPAR defines a new parameter compensator. The value "surf" is the surface number in the original lens file. Zemax automatically renumbers the surface if additional coordinate break or other dummy surfaces were inserted by the tolerance program. The value for "param" corresponds to the parameter number for the surface.

FORMAT

Purpose:
Specifies the numerical precision format for subsequent REPORT statements.

Syntax:
```
FORMAT m.n [EXP]
```

Discussion:
The integers m and n are separated by a decimal point. The value m refers to the total number of characters to be printed, even though some of them may be blank. The value n refers to the number of places to display after the decimal point. Therefore, FORMAT 8.4 will cause subsequent REPORT statements to print 8 characters, with 4 numbers after the decimal point. FORMAT .5 will cause REPORT to show 5 decimal places, and as many total places as needed. FORMAT only affects numeric output from REPORT. If a number is too large to fit within the m decimal places, then the m portion of the FORMAT statement will be ignored. The optional keyword EXP after the m.n expression indicates exponential notation should be used. The default format is 16.8 EXP.

Example:
```
FORMAT 18.9 EXP
```

GETMERIT

Purpose:
Evaluates the current merit function.

Syntax:
```
GETMERIT
```

Discussion:
GETMERIT should be used prior to any REPORT commands if the merit function has not yet been evaluated. LOADMERIT, OPTIMIZE, and OPTIMIZE-OD also update the current merit function.

HEADER

Purpose:

Prints a text header to the beginning of an output file. The header can be used to identify to which tolerance operand or Monte Carlo file the subsequent data belongs to.

Syntax:
```
HEADER "text"
```

Discussion:
The HEADER command adds a text line to the output file when using OPENFILE in TXT mode. The HEADER command will output the user supplied text string, followed by either the tolerance operand number (during sensitivity analysis) or the Monte Carlo file number being evaluated, or an indication that the nominal performance is being evaluated. For sensitivity analysis the min is done first, then the max.

LOADMERIT

Purpose:
Loads a new merit function to use as the tolerance criterion.

Syntax:
```
LOADMERIT filename.mf
LOADMERIT "C:\SOMEPATH\SOME MERIT FILE NAME.MF"
```

Discussion:

The merit function file must reside in the <data>\MeritFunction folder (for details, search the help files for "Folders") if no path is specified. The file should be in the proper format as saved using the Toolbar option of the Merit Function Editor, or may be a full ZMX file, in which case only the merit function portion of the file is read and loaded.

When the merit function is loaded, it replaces any existing merit function. Any operands in the merit function which reference surface numbers should be the surface numbers for the original, unaltered lens. Zemax automatically adjusts the surface numbers as required if additional coordinate break or other dummy surfaces were inserted by the tolerance program. The new merit function is then evaluated. The resulting merit function value is what will be returned to the tolerance analysis program as the criterion unless a subsequent GETMERIT, LOADMERIT, OPTIMIZE, or OPTIMIZE-OD command is executed.

NOMINAL

Purpose:
Allows some control over how the nominal performance is evaluated.

Syntax:
```
NOMINAL keyword
```

Discussion:
The NOMINAL script command allows some control over the evaluation of the script for the computation of the nominal performance only. The supported keywords are:
OPT_OFF: Turns off optimization of the compensators. Any subsequent OPTIMIZE or OPTIMIZE-OD com- mands are ignored.
OPT_ON: Turns on optimization of the compensators. Any subsequent OPTIMIZE or OPTIMIZE-OD com- mands are executed.
The NOMINAL command is only executed during computation of the nominal system performance, and not during the analysis of individual tolerances.

OPENFILE

Purpose:
Opens a text or binary file for subsequent output of data using REPORT.

Syntax:
```
OPENFILE "FILENAME" mode
```

Discussion:

FILENAME may be any valid file name, with the full path and extension included, such as "C:\DATA\My File.DAT". The mode parameter should be BIN for a binary file, or TXT for a text file. OPENFILE will open a new data file the first time the script is run for a tolerance analysis run. Each subsequent call to the script will append data to the same file.

Data is written to the file whenever the REPORT OR HEADER commands are executed. However, the text string is not included in REPORT commands; just the numerical data the REPORT command prints to the tolerance output file. If mode is TXT, the file will be text with each REPORT value written on a separate line. If mode is BIN, each value will be written out as a 64-bit double precision number.

Because data is written out each time the script is called, the amount of data in the file depends upon how many tolerance operands and Monte Carlo runs are being performed. The first call to the script is to compute the nominal criterion, then the sensitivity analysis, then the Monte Carlo analysis. OPENFILE should not be used during an inverse sensitivity run, because there are multiple calls to execute the script and no practical way to discern which data belongs to which operand.

The file is automatically closed after the script is finished executing. Multiple OPENFILE commands may be used within the same file to write different data to different files. See also CLOSEFILE.

OPTIMIZE

Purpose:
Optimizes the system using the current merit function and compensators.

Syntax:
```
OPTIMIZE n
```

Discussion:
OPTIMIZE calls the damped least squared optimizer and executes "n" cycles of optimization. If n is zero or is omitted the optimizer runs in automatic mode, terminating when convergence is detected.

OPTIMIZE-OD

Purpose:
Optimizes the system using the current merit function and compensators.

Syntax:
```
OPTIMIZE-OD n
```

Discussion:
OPTIMIZE-OD calls the orthogonal descent optimizer and executes "n" cycles of optimization. If n is zero or is omitted the optimizer runs in automatic mode, terminating when convergence is detected.

PERTURB

Purpose:
Randomly perturbs a parameter in the lens.

Syntax:
```
PERTURB type int1 int2 int3 stat nstd min max
```

Discussion:
PERTURB is used to randomly change a numerical parameter in the lens. The integer parameter type must be 0 for surface data, 1 for parameter data, 2 for extra data, 3 for multi-configuration data, 4 for non-sequential component position data, and 5 for non-sequential parameter data. The arguments to PERTURB change meaning depending upon the value of type as described below.

Type = 0, surface data: int1 is used for the surface number, int2 is 0 for thickness, 1 for radius, 2 for conic, and 3 for semi-diameter. The value of int3 must be set to zero.

Type = 1, parameter data: int1 is the surface number, int2 is the parameter number, int3 must be set to zero.

Type = 2, extra data: int1 is the surface number, int2 is the parameter number, int3 must be set to zero.

Type = 3, multi-configuration data: int1 is the configuration number, int2 is the operand number, int3 must be set to zero.

Type = 4, NSC position data: int1 is the surface number, int2 is the object number, int3 must be 1 for x, 2 for y, 3 for z, 4 for tilt x, 5 for tilt y, or 6 for tilt z.

Type = 5, NSC parameter data: int1 is the surface number, int2 is the object number, int3 is the parameter number.

The stat integer indicates the statistical model to use: 0 for normal (Gaussian), 1 for uniform, and 2 for parabolic. Only these 3 distributions are currently supported. The nstd is a floating point value that indicates the number of standard deviations from the center of the distribution to either edge. A small value for nstd (less than 1.0) will yield a nearly uniform distribution. Large values of nstd will yield a sharply peaked distribution, with most of the randomly generated values being clustered close to the center of the range. Only the normal distribution uses the nstd value, however, it must be provided and should be set to zero if the other stat types are used.

The min and max values indicate the range over which the random values are selected.

For example, to randomly perturb the thickness of surface 6 by -0 to +1 with a normal distribution covering plus/minus 3 standard deviations, the syntax would be:

```
PERTURB 0 6 0 0 0 3.0 0.0 1.0
```

The resulting random value will be between 0.0 and 1.0, and will likely be close to 0.5 because the number of standard deviations is moderately large. This random value will then be added to the current value for the thickness of surface 6. Note that PERTURB works on all values, whether they are compensators or not. To perturb values controlled by multi-configuration operands, use type 3 to perturb the operand directly rather than the parameter under control.

The PERTURB command is ignored while evaluating the nominal criterion.

QUIET

Purpose:
Controls whether data sent to a file using REPORT is also sent to the tolerance analysis output window..

Syntax:

```
QUIET ON
QUIET OFF
```

Discussion:

The default setting for QUIET is OFF. If QUIET is OFF, then the output of all REPORT commands is sent to the tolerance analysis window. If QUIET is ON, then the output of REPORT commands will not be sent to the tolerance analysis window. Any output will still be sent to any files opened with OUTPUTFILE, however.

REPORT

Purpose:
Send numerical or text data to an output file.

Syntax:
```
REPORT "text" operand
REPORT "text" SZERNIKE term field wave sampling maxorder
REPORT "text" -1
REPORT_NB (see discussion under syntax 4 below)
```

Discussion:
There are different ways to use the REPORT or REPORT_NB commands:
Syntax 1: REPORT will print any user defined text to the tolerance output window, along with the value of any operand in the currently loaded merit function. The value for operand is an integer corresponding to the operand number (row) of the value to print (note that any value may be computed in a merit function, and if it is not needed for optimization, it may be weighted to zero and is still available for reporting). If operand is zero, then the value of the entire merit function will be printed out. The numeric format is set by the FORMAT command.
Syntax 2: REPORT, when followed by a text string and then the keyword SZERNIKE, can be used to compute and report Standard Zernike coefficients (for details, search the help files for "Zernike Standard Coefficients"). If the term number is zero, then the other values for field, wave, sampling, and maxorder are used to define the Zernikes to be computed. The computed values are then saved in a buffer, and no output is generated. If term is between 1 and the most recent maxorder value used in a previous SZERNIKE call, then that Zernike term is reported. A sample session follows:

```
REPORT "Compute Only!" SZERNIKE 0 1 1 1 4
REPORT "Zernike 1 = " SZERNIKE 1
REPORT "Zernike 2 = " SZERNIKE 2
REPORT "Zernike 3 = " SZERNIKE 3
REPORT "Zernike 4 = " SZERNIKE 4
```

Note that only when the term is zero are the Zernikes actually computed; the following calls retrieve the computed Zernikes one at a time.

Syntax 3: REPORT, when followed by a text string in quotes and then the integer -1, can be used to print any text string (or blank line if the string is empty).

Syntax 4: If the command REPORT_NB is used instead of REPORT, then no line break is printed at the end of the line, so multiple data may be displayed on one line. The last report issued should use REPORT and not REPORT_NB to properly terminate the line, otherwise the last REPORT_NB value will be at the end of a Monte Carlo run. The remaining arguments are supported as described for syntax 1, 2, and 3.

See also "HEADER" and "QUIET", and "OPENFILE".

SAVE

Purpose:

Saves the current lens file.

Syntax:
```
SAVE n
```

Discussion:
SAVE will save the current lens file in a ZMX file; with the name TSAVnnnn.ZMX where nnnn is the four digit integer representation of the number "n". For example, if "n" is 6, the file will be saved as TSAV0006.ZMX.

UPDATE

Purpose:
Updates the current lens.

Syntax:
```
UPDATE
```

Discussion:
Updates the current lens. Use after PERTURB commands.

Tolerance Script example

As an example, suppose an optical system is assembled from multiple lens elements. As part of the alignment and evaluation process the system is adjusted as follows:

Element number 2 is decentered until an axial test beam is centered on the image.

Element number 4 is then shifted along the axis until the proper magnification is achieved.

Finally, the back focus is adjusted to maximize the on-axis MTF only.

The distortion is then measured and logged.

The MTF at 5 field points is evaluated.

Assume that element number 2 is decentered using parameters 1 and 2 of surface 3, which is an existing coordinate break, the position of element 4 is thickness 10, and the back focus is thickness 15. Furthermore, assume a merit function which centers the axis ray on the image surface is defined in CENTER.MF, the magnification enforcing merit function is MAGNIFY.MF, the MTF merit function is MTF.MF, and the evaluation merit function is EVALUATION.MF. The corresponding script might look like this:

```
! clear any existing compensators for a clean start
CLEARCOMP
! load the centering merit function
LOADMERIT CENTER.MF
! define the two compensators to decenter element 2
CPAR 3 1
CPAR 3 2
! optimize 4 cycles
OPTIMIZE 4
! clear the decenters, load the magnification merit, and adjust thickness 10
CLEARCOMP
LOADMERIT MAGNIFY.MF
COMP 10 0
```

```
OPTIMIZE 4
! now load the MTF merit function, and adjust back focus
CLEARCOMP
LOADMERIT MTF.MF
COMP 15 0
OPTIMIZE 4
! load the evaluation merit function, report the distortion and 5 MTF values
! these should be the first 6 operands in EVALUATION.MF.
CLEARCOMP
LOADMERIT EVALUATION.MF
REPORT "Distortion    = " 1
REPORT "MTF at field 1 = " 2
REPORT "MTF at field 2 = " 3
REPORT "MTF at field 3 = " 4
REPORT "MTF at field 4 = " 5
REPORT "MTF at field 5 = " 6
```

At the end of the script, the entire merit function value resulting from the last LOADMERIT or OPTIMIZE command is returned as the criterion which Zemax reports and uses.

Tolerancing multi-configuration (zoom) lenses

Tolerance analysis can be performed on all or any specific configuration of a multiple configuration lens. Select the desired configuration from the "Configuration #" drop down box. When using inverse tolerances, successive tolerance analysis in each configuration individually will yield the tightest tolerances which apply to all configurations collectively.

In some cases, Zemax cannot perform tolerances correctly on surfaces controlled by multi-configuration operands. For example, consider an element decenter defined by the tolerance operand TEDX 3 4, while the radii and thicknesses of these surfaces are also being controlled by thermal operands in the multi-configuration editor. In this case Zemax will issue an error message. The solution is to separate the two effects by adding dummy surfaces before and after the element. The new dummy surfaces (3 and 6) are added before and after the real lens surfaces (now numbered 4 and 5). Now the decenter can be specified using TEDX 3 6, and the thermal controls may still be applied to surfaces 4 and 5.

Tolerancing with solves

Generally speaking, delete all ray-based solves before tolerancing a lens. Ray based solves, such as marginal ray height solves, no longer have meaning when the optical system contains tilted or decentered elements. Even if the system is initially rotationally symmetric, most tolerance commands such as TETX and TETY force the system to be non-rotationally symmetric. Curvature solves, such as the F/# solve, would dynamically alter the curvature of a surface during tolerancing, which does not reflect the behavior of the real system.

There are occasions where a pickup solve is required, and thus should not be deleted. For example, if the lens is used in double-pass, and a tilt on one element implies a tilt on a subsequent element, then a pickup solve can be used to pickup the tilts from the first element. For this case use the TUTX and TUTY tolerance operands.

Trouble shooting the tolerance results

See the discussion "SAVE".

If any of the calculated tolerance data show a value of "Infinity", this means the criterion could not be evaluated for the specified tolerance. Usually the criterion cannot be evaluated because of total internal reflection of some of the ray targets. The statistical data which follows the sensitivity analysis is usually meaningless if any of the tolerances have a criterion value of infinity. One solution is to decrease by a factor of two or more the tolerance value, and then repeat the analysis.

Optimizing for tolerance sensitivity

See "Optimizing tolerance sensitivity".

Pitfalls when tolerancing

One possible error is non-physical propagation of rays when using tilt tolerances such as TETX and TETY. If two elements are separated by a very small air space or dummy surface, the default tolerances will include tilting of each element independently. If the element spacing is small, it is possible for one of the elements to be tipped to the point where it "collides" with the other element. This could not happen in practice, however, for small amounts of tilt the tolerance values are still a good indication of performance.

If the nominal criterion the tolerance algorithm reports is different from the expected value, check to see if ray aiming is "off" in the lens file (search the help files for "Ray Aiming" for details about ray aiming). Try using ray aiming "on" and re-optimizing. Generally, if there is a big difference between criterion when ray aiming is turned on and off, then it should be left on.

Summary

The tolerance routine is quite flexible and powerful. Zemax uses no approximations, extrapolations, or estimations to compute the tolerances. For this reason, it gives useful results for both conventional and complex systems. It is very important to appreciate that tolerancing is a complex procedure, and the algorithms used by Zemax to manipulate the lens data are not infallible. Therefore, it is the designer's responsibility to verify that the program is computing reasonable results.

Using Part Designer

The Zemax Part Designer (ZPD) is an interface that allows users to create and manipulate geometries in an environment separate from Zemax. Special features such as the "Ghost View" and On-The-Go Geometry creation allow users to quickly build, assess, and modify individual steps leading up to the final object. Using the ZPD console, users may quickly modify and add commands and operations to a geometry model and generate results within moments. Overall, the Part Designer allows for a streamlined process to incorporate solid 3-D geometry into the user's Zemax non-sequential designs.

Getting Started

For Details on the user interface, see Part Designer Tab
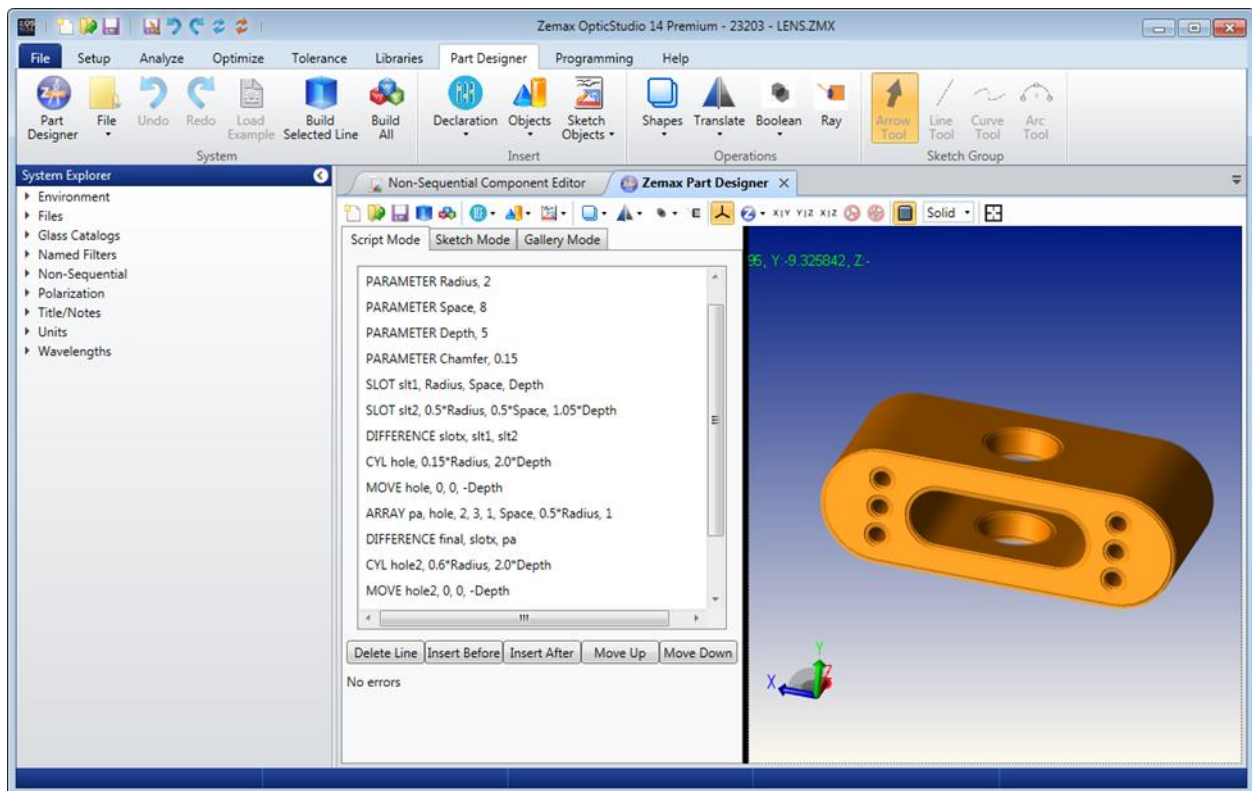
ZPOs and ZSOs

Objects utilizing gallery data and sketches created in the Gallery Mode and Sketch Mode's, respectively, must be saved using the extension, ZPO. The ZSO file is a readable text file only capable of saving a script. The ZPO in many ways is much more robust, allowing for the creation of more complicated parts with "Sketch Object" commands. ZPO's may also be saved with Gallery Data (accessed in "File"-> "Save Gallery Data"). This information is user defined and is optional. ZSO's and ZPO's without gallery data will not be exposed in the gallery mode viewer. Saving new parts with the ZPO extension is recommended even if gallery and sketch data are not needed.

Part Creation

The console acts as a text editor in a coding environment. Each line of the console can accept one command.

The syntax for the ZPD script is as follows:

- Declaration commands. To define numerical values used elsewhere in the script the general syntax is:
  ```
  declaration new_name, value
  ```

- Object creation commands. To create a new object, the general syntax is:
  ```
  object_type_name new_object_name, arguments
  ```

- Operation commands. To operate on an existing object, the general syntax is:
  ```
  operation_type_name existing_object_name, arguments
  ```

- Computation commands. To compute data about an existing object, the general syntax is:
  ```
  computation_type_name new_data_name, arguments
  ```

The Zemax Part Designer allows for commented lines by placing an exclamation point in front of the commented line. Like in any coding environment, Zemax Part Designer also allows users to perform calculations within a command. For example:

```
PARAMETER radius, 10
CONSTANT radius2, (radius*0.5)+1
```

These calculations may also be used in the creation of all geometries and within the "Insert" dialog options.

Building a Part Object

The user may build a part at any point as long as the part contains at least one object creation script command. To build a part, press the "Build…" button on the Script Mode toolbar or navigate to File->Rebuild (Ctrl+G). A progress bar will spawn to indicate which line of code the Zemax Part Designer is tessellating. The user may not perform any actions with Zemax Part Designer at this time. Please note that some complex commands (e.g. Array, Ring, Spiral, etc.) will take longer to tessellate. If there are any errors within the script, ZPD will spawn a popup window indicating the type of error and line in which it is located. Common errors are detailed as follows:

- "Parameter/Constant/Object at line x has already been defined": The parameter or constant has been defined more than once.

- "Too many parameters/constants/objects at line x": More than one value has been listed to define a parameter or constant, or too many objects are used to define commands such as Boolean operations or Shape operations.

- "Error in Tessellation at line x": The Part Object could not be tessellated. This may be caused by certain values within a command line exceeding constraints or conflicting with one another.

- Geometry Specific Errors: Generally, tessellation errors are a result of faulty parameters. For example, a volume cannot have a depth or length of 0. The geometry specific errors will notify users of which line, and which set of parameters

- Various Syntax Errors

If an error occurs, ZPD will tessellate all objects and operations within the script up to the error line.

Scripting Syntax and Math Syntax
Objects can be built and manipulated via the commands in the Part Designer Tab, or they can be scripted into text editor in the Script Mode window. See System Group, Insert Group, and Operations Group for details on each of the commands.

Creating a Sketch

Sketches are used in conjunction with the commands listed within the "User Defined" subsection of the commands.

There are several guidelines for creating sketches:

- Objects utilizing sketches must be saved with the .ZPO extension.

- All sketches exist on the X-Y plane. The EXTRUDE and PYRAMID_S commands will extend along the +Z axis. The REVOLVE command will always revolve the sketch about the Y-axis.

- A "curve" is defined as an array of connected "segments". Each segment is defined by 2 consecutively numbered points. The first point of any curve will always be point 1, the next will be point 2, and so on.

- The ordering of points and direction in which segments are drawn has significance. The notation for point indices and their relationships to adjacent segments is shown below:
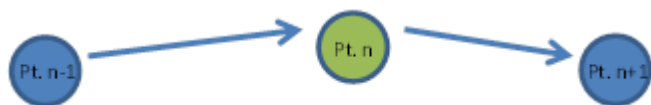
The point highlighted in green represents the user's selected point with the index n. The directions of the arrows indicate the order in which the user has drawn the curve; in this case, the user has drawn the figure from left to right.

- Sketches used with REVOLVE may exist on the axis of rotation (Y-Axis), but must occupy only one side of the axis.
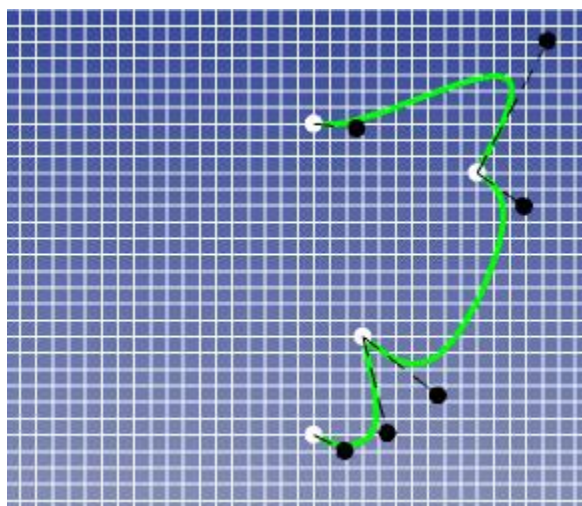
The sketch can only exist on one side of the axis of rotation as shown above.

- Sketches used with the EXTRUDE, PYRAMID_S, and REVOLVE commands must be closed. This means that the last point specified by the user must be the same as the starting point of the sketch. Sketches will automatically close when a point close enough to the starting point is specified.
- Sketches may consist of both straight line segments, Bezier curves, and arc segments. A Bezier curve may be formed into a line segment, but a line segment cannot be adjusted to form a smooth curve unless deleted and replaced with a curve.

-

Using a Sketch

Once a sketch is completed, it may be selected from any of the "Sketch Object" command options. Navigate out of sketch mode and into the Script Mode. The Sketch ID of a command indicates the name of any existing sketches. Incomplete sketches (not closed) cannot be used for the EXTRUDE, REVOLVE, or PYRAMID_S commands.
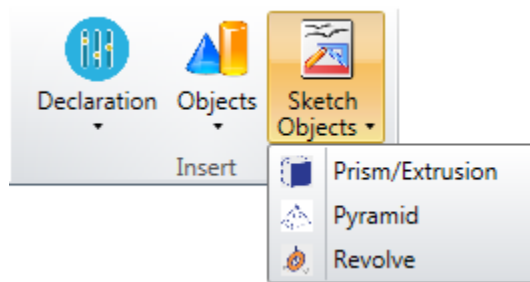
What is a Bezier Segment?

The parametric Bezier Segment is often used within computer graphics to draw smooth, organic lines. The Zemax Part Designer utilizes Cubic Bezier Segment, meaning it has a total of 4 control points consisting of the start and end points of the curve as well as 2 points (referred to as tangent points) defining the control polygon dictating the curves shape. These tangent points are shown below as P1 and P2.
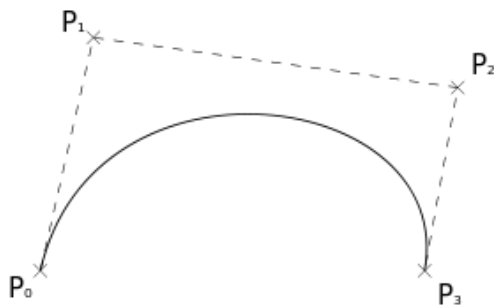
A Bezier Segment shown with four control points. P1 and P2 are used to control the direction of the segment at each end point.

The equation for a cubic Bezier segment is:

$$[x, y] = (1 - t)^3 P_0 + 3(1 - t)^2 t P_1 + 3(1 - t)t^2 P_2 + t^3 P_3$$

Where [x,y] are points along the segment evaluated from t = 0 to t = 1. P¬0 is the "start" point of a segment, P_1 is the "start tangent" point of a segment, P_2 is the "end tangent" point of a segment, and P3 is the "end" point of a segment.

This manual will often reference "continuous" intersections between various segments. When two adjacent segments are "continuous" or "smooth", there exists, at minimum, a first order parametric continuity (C^1 continuous) on the curve at the intersection between the two segments. However, a smooth curve does not entail a second order parametric continuity (C^2 continuous) in which the first AND second derivatives are the same. To demonstrate this visually, refer to the figures below:
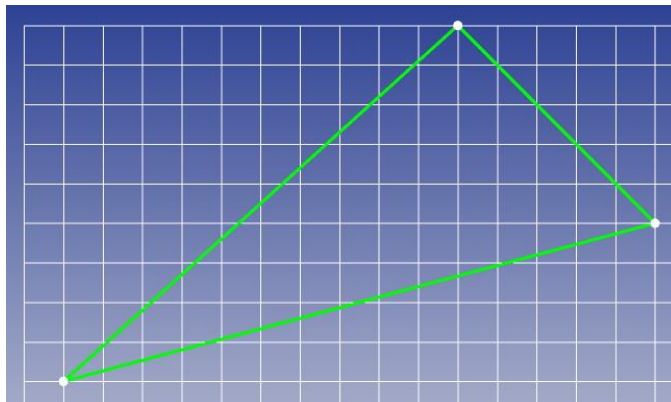
A C^0 continuous curve. In other words, the points are connected but lack and first or second order continuity

A C2 or C1 continuous curve can be easily constructed with a combination of Bezier, line, or arc segment. Below, two Bezier segments are linked to form a C2 continuous curve.
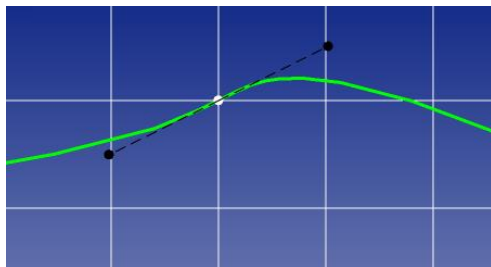
A C2 continuous curve. In a C1 and C2 continuous curve, the end tangent point of the previous segment, the end/start point of the previous/next segment, and the start tangent of the next segment are all collinear. In a first order continuous curve, the line segment formed by the previous segment's end tangent point and end point may not be of the same magnitude of the line segment formed by the next segment's start tangent point and start point. However, they will be of the same direction.

This manual and Help files will often detail the manipulation of the start and end tangent points of various segments. The straight line segment formed by connecting the start tangent point with the start point of a segment or the end

tangent point with its respective end point is indicated with a stippled line in ZPD. Each Bezier Segment has a start and an end tangent point associated with the segment's start and end control points. We will call the line segment formed by the tangent points to their respective end or start points as the end or start tangent vectors.

Exposing and Parameterizing Sketches within Zemax

The characteristics of a sketch can be easily parameterized for use within Zemax. This feature is instrumental in optimizing Part Designer Objects for use inside an optical system. Selecting a point within the sketch editor will display the point manipulation panel.

The point manipulation panel allows users to alter the x and y coordinates as well as the tangent points appended to the selected point; however, modifiable tangent points will only appear when the curve preceding or following the point is a Bezier curve. Users may choose to use the mouse to manipulate these points or input them into the text fields of the point manipulation panel. Pressing the "Set" button will enact these changes.

By checking the "Expose Point" box, the point's coordinates as well as the tangent coordinates associated with the point will all be exposed within the Zemax Environment.

Constraining a Curve

Line, Bezier, and Arc segments can also be set tangent to one another to create a "smooth curve". The effects of checking the "Smooth Curve" box of the point manipulation panel are shown below.

The point manipulation panel for Bezier Curves and line segments

When a curve is continuous, the tangent points and the control point will be collinear. Adjusting one tangent point will automatically adjust the other tangent point, thus maintaining a smooth curve. Enabling smoothing between a line segment and a Bezier segment will render the segment parallel to the vector between the end point and the tangent point.

The following table indicates the effects of adding a smoothing constraint to a point joining all combinations of segment types. The table also details the behavior of the curve upon altering the selected point.

Changes made to sketch parameters that are exposed in Zemax will follow the procedure taken by the **Point Control Panel** actions detailed in the table below.

Smoothing Constrain Behavior

		Next Segment		
		Line	Bezier	Arc
Previous Segment	Line	**N/A: Smoothing disabled**	**Initial:** As the smoothing constraint is applied to this point, the line segment preceding the selected point is unchanged. The following Bezier Segment aligns itself with the line segment by moving its start tangent point a distance of x units opposite the direction of the line segment, where x is the length of the line segment. **Point Control Panel:** Altering the point coordinates after the constraint has been set will adjust the start point of the Bezier Segment and end point of the line segment. Meanwhile, the Bezier segment's start tangent vector will behave in the same manner as initially setting the constraint by aligning itself with the line segment. Utilizing the Point Control Panel to modify the start tangent point will adjust the direction of the previous line segment so it points in the opposite direction of the start tangent vector. Any changes made to the following segment will be propagated down the rest of the curve. **Mouse Movement:** Left-click dragging the selected point will alter the segments in the same manner as altering the [x,y] coordinates in the point control panel. Right-click dragging the start tangent point will alter magnitude of the start tangent vector; however, the direction between the two points will not change.	**Initial:** The line segment prior to the arc segment will maintain its length and align itself with the start direction of the arc. The direction of the arc is represented by a unit vector originating at the start point of the arc while laying tangent to the arc at the start point. **Point Control Panel:** Altering the point coordinates of the selected point will adjust the radius of the arc segment as well as the start point of the arc. Meanwhile, the previous line segment will maintain its length and adjust its slope to match the start direction of the arc. Modifying the start point of the prior line segment will adjust the start direction and radius of the arc segment accordingly. **Mouse Movement:** Adjusting the control points will yield the same results as point control panel modification. **Arc Control Panel:** Changing the radius of the arc segment will not change the location of the arc's control points; however, altering the radius will change the start and end direction of the arc and will therefore adjust the prior line segment's slope to account for this.

		Initial:	Initial:	Initial:
		As the smoothing constraint is applied to this point, the line segment following the selected point is unchanged. The previous Bezier Segment aligns itself with the line segment by moving its end tangent point a distance of x units opposite the direction of the line segment, where x is the length of the line segment.	As the smoothing constraint is applied to the selected point, the preceding Bezier segment is unchanged. The following Bezier segment has the start tangent vector's magnitude is set to a length of x, where x is length of the prior Bezier segment's end tangent vector. The direction of the next segment's start tangent vector becomes opposite to the prior curve's end tangent vector, thus creating a G^2 continuous (first and second derivatives are continuous) curve at the point of intersection.	Applying the smoothing constraint will not change the arc segment. The preceding Bezier segment will align its end tangent vector to match the start direction of the arc. The magnitude of the end tangent vector will not change.

Bezier (rotated row label)

Initial:
As the smoothing constraint is applied to this point, the line segment following the selected point is unchanged. The previous Bezier Segment aligns itself with the line segment by moving its end tangent point a distance of x units opposite the direction of the line segment, where x is the length of the line segment.

Point Control Panel:
Altering the point coordinates after the constraint has been set will adjust the end point of the Bezier Segment and start point of the line segment. Meanwhile, the Bezier segment's end tangent vector will behave in the same manner as initially setting the constraint by aligning itself with the line segment.

Utilizing the Point Control Panel to modify the end tangent point will adjust the direction of the next line segment so it points in the opposite direction of the end tangent vector. Any changes made to the following segment will be propagated down the rest of the curve.

Mouse Movement:
Left-click dragging the selected point will alter the segments in the same manner as altering the [x,y] coordinates in the point control panel.

Right-click dragging the end tangent point will alter magnitude of the end tangent vector; however, the direction between the two points will not change.

Initial:
As the smoothing constraint is applied to the selected point, the preceding Bezier segment is unchanged. The following Bezier segment has the start tangent vector's magnitude is set to a length of x, where x is length of the prior Bezier segment's end tangent vector. The direction of the next segment's start tangent vector becomes opposite to the prior curve's end tangent vector, thus creating a G^2 continuous (first and second derivatives are continuous) curve at the point of intersection.

Point Control Panel:
Modifying the selected point's [x,y] coordinates via the point control panel will only adjust the end/start points of the previous and following segments. The start and end tangent points will be offset according to the new location, however the direction and magnitude of the tangent vectors will not change.

If a smooth point consisting of two Bezier Segments is selected, the user may only alter the value indicated by the previous segment's end tangent points. Adjusting the end tangent point will not alter the location of the end or start points, but will move the start tangent point of the next segment to maintain a G^2 continuous curve at the point.

Mouse Movement:
The respective mouse actions for moving points and tangent points will yield the same results as mentioned above.

Initial:
Applying the smoothing constraint will not change the arc segment. The preceding Bezier segment will align its end tangent vector to match the start direction of the arc. The magnitude of the end tangent vector will not change.

Point Control Panel:
Adjusting the selected point's [x,y] coordinates will move the arc's start point and affect its radius. The end point of the preceding Bezier segment follows the changes. The end tangent vector is adjusted to align with the arc segment's start direction. The magnitude of the end tangent vector is unaffected.

Modifying the end tangent vector of the preceding curve will align the following arc segment's start direction to the altered end tangent vector. As a result, the arc's start point and radius may also change.

Mouse Movement:
Left-clicking the selected point and dragging it will adjust the start point and radius of the next arc segment. The end tangent vector of the preceding Bezier segment will align itself with the changing start direction of the arc to maintain first order continuity.

Right-clicking and dragging the end tangent vector will increase or decrease its magnitude, but will not change its direction.

Arc Control Panel:
Changing the radius of the arc will alter its start and end directions, forcing the end tangent vector of the previous curve to adjust accordingly to maintain continuity. The magnitude of the end tangent vector does not change.

		Initial: The line segment following the arc segment will maintain its length and align itself with the end direction of the arc. The direction of the arc is represented by a unit vector originating at the end point of the arc while laying tangent to the arc at the end point.	**Initial:** Applying the smoothing constraint will not change the arc segment. The following Bezier segment will align its start tangent vector to match the end direction of the arc. The magnitude of the start tangent vector will not change.	**Initial:** When the smoothing constraint is applied between two arc segments, the arc segment prior to the selected point will remain unchanged. The arc segment following the selected point will alter its start direction to align itself with the end direction of the prior arc resulting in a change to the radius.
		Point Control Panel: Altering the point coordinates of the selected point will adjust the radius of the arc segment as well as the end point of the arc. Meanwhile, the following line segment will maintain its length and adjust its slope to match the end direction of the arc.	**Point Control Panel:** Adjusting the selected point's [x,y] coordinates will move the arc's end point and affect its radius. The start point of the preceding Bezier segment follows the changes. The start tangent vector is adjusted to align with the arc segment's end direction. The magnitude of the start tangent vector is unaffected.	**Point Control Panel:** Changing the [x,y] coordinates of the selected point will set the prior arc segment's end point to the specified coordinates, while adjusting the radius to account for this change. The following arc segment will utilize the new end direction of the prior arc and adjust its own start direction, start point, and radius to compensate for these changes.
		Modifying the end point of the following line segment will adjust the end direction and radius of the arc segment accordingly.	Modifying the start tangent vector of the following segment will align the preceding arc segment's end direction to the altered start tangent vector. As a result, the arc's end point and radius may also change.	**Mouse Movement:** Left-clicking and dragging the selected point will yield the same results as point control panel modification.
		Mouse Movement: Adjusting the control points will yield the same results as point control panel modification.	**Mouse Movement:** Left-clicking the selected point and dragging it will adjust the end point and radius of the preceding arc segment. The start tangent vector of the next Bezier segment will align itself with the changing end direction of the arc to maintain first order continuity.	**Arc Control Panel:** Making changes to an individual arc's start point, end point, or radius will propagate the change to adjacent segments. For example: If the user modifies the radius of the previous arc, the new end direction will alter the next arc so that the next arc's start direction, radius, and start point adjust to the modifications made to the prior arc.
	Arc	**Arc Control Panel:** Changing the radius of the arc segment will not change the location of the arc's control points; however, altering the radius will change the start and end direction of the arc and will therefore adjust the following line segment's slope to account for this.	Right-clicking and dragging the start tangent vector will increase or decrease its magnitude, but will not change its direction. **Arc Control Panel:** Changing the radius of the arc will alter its start and end directions, forcing the start tangent vector of the previous curve to adjust accordingly to maintain continuity. The magnitude of the start tangent vector does not change.	

Naming Points and Arcs

ZPD allows users to rename points and arcs. As a point created, ZPD will allot points a name using the format "<Sketch #.Point #>". Arcs follow a similar convention using the format "<Sketch #.Arc #>". Using the "Rename" button on the Arc and Point Control Panels, users may give the object any name that has not yet been utilized.
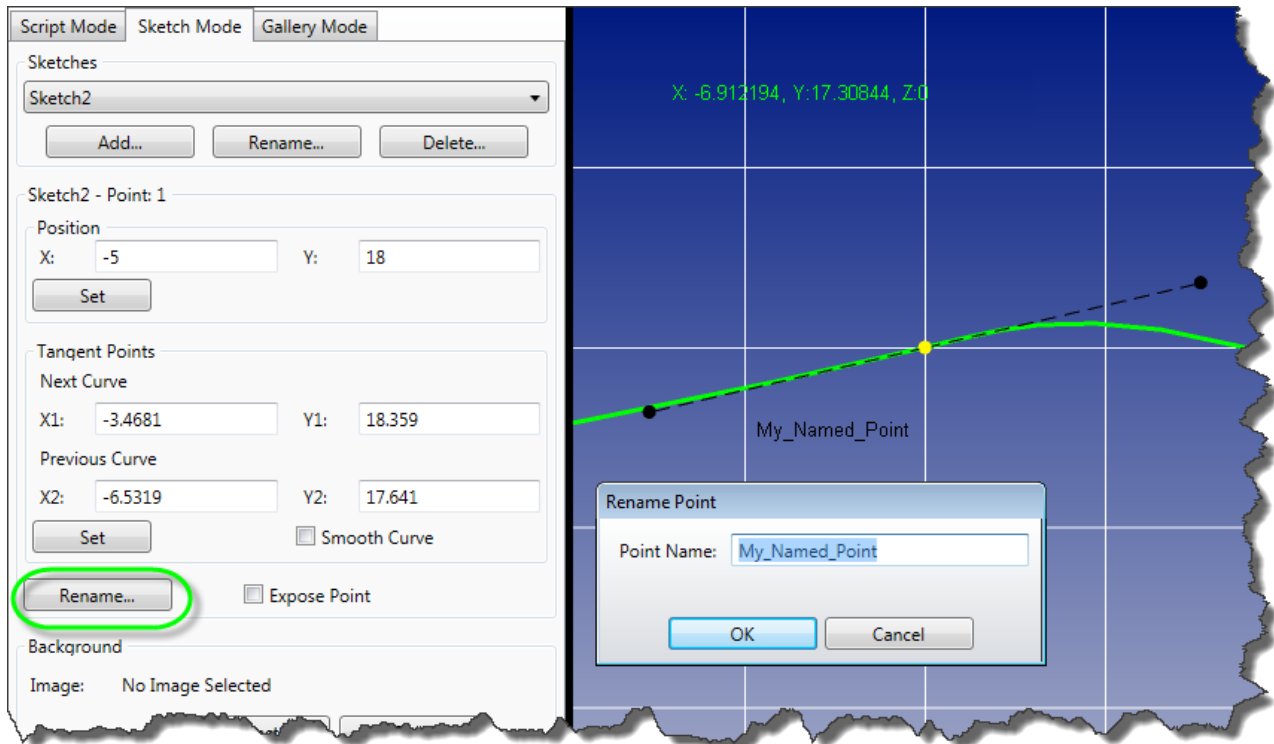
After a point has been renamed, red lettering of the given name will appear next to the newly named point. Renaming an arc will place the red text at the center point of an arc segment.

Upon exposing the point or arc, the new name will appear in the Zemax editor when the part is loaded.
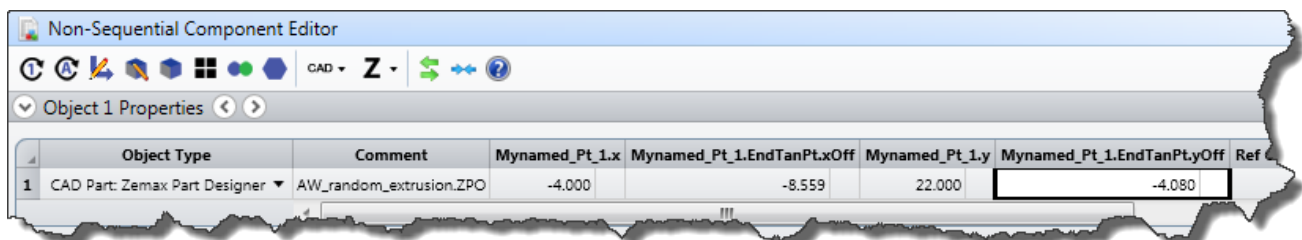
An exposed point with the name "Mynamed_Pt_1" is modified within the Zemax Editor.

Gallery Mode

The Gallery Mode provides a view of all available ZPO files located within the default Zemax Part Designer directory. Any file created within the Zemax Part Designer can store gallery data.

To create gallery data for an object, it must be saved using the Save Gallery feature. This function saves the file in its current state along with a user description and bitmap image.
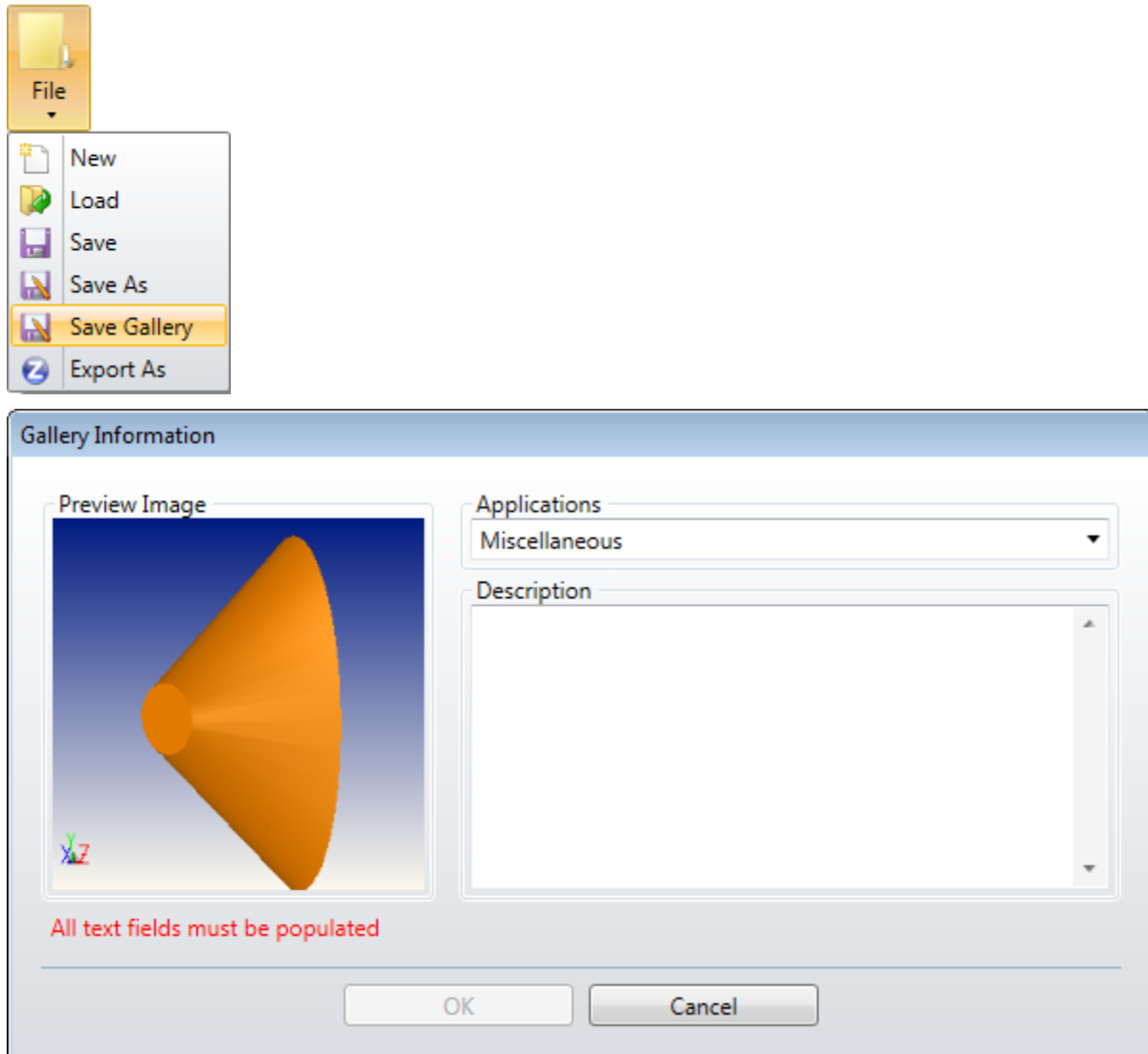
The Gallery Information window opens upon selecting the "Save Gallery" feature. This window shows the bitmap image that will be saved. The application chosen for the part will determine how it is sorted in the Gallery Mode. A description of the object must also be entered prior to saving.

Information inputted in the "Applications" drop down menu can be specified by the user or chosen from the available items. Items that do not correspond to the listed items will automatically be classified under "Miscellaneous". When taking a screenshot of the part, the Zemax Part Designer utilizes a square in the middle of the viewing window of height and width equivalent to the current height of the viewing window. Press "OK" to save the gallery mode information. All data fields and screenshots must be entered before saving. The part must be saved with the .ZPO extension.

Commands

ZPD Commands fall into several categories:

- Insert Group Commands: Create new objects or variables for use in creating complex Zemax Part Objects.

  - Declaration Commands: Used to store variables for use in Zemax and within the script

  - Object Creation Commands: The building blocks of an object, these features create geometries

  - Sketch Object Commands: Create objects from sketches

- Operation Group Commands: Allow objects created by Insert Group commands to be modified and interact with one another

  - Shapes Commands: Duplicate or modify existing objects.

  - Translate Commands: Move existing objects

  - Boolean Commands: Create a new object using Boolean logic and existing objects

  - Ray Command: retrieve quantitative values from the geometry to be used as parameters and constants in other calculations.

  - Math Syntax: The Zemax Part Designer is also capable of performing simple mathematical calculations as a substitute for data values within a script

Insert Group

Declaration Commands

Used to store variables for use in Zemax and within the script

The declaration commands are used to define numerical values that may be used elsewhere in the script by reference to the declared value name. There are two types of declarations: parameters and constants. The difference between a parameter and a constant is that a parameter is exposed to the Zemax user interface, and the value defined in the script may be overridden with a different value in the non-sequential component editor.

Parameter

PARAMETER name, value
The PARAMETER command defines the name and numerical value of a constant which can subsequently be used in the place of any numerical value. This named parameter is also automatically exposed to Zemax as a user definable parameter. Changing or optimizing the parameter in Zemax will rebuild the entire object with the new value, and any value defined in Zemax will override the value in the script. The script value will only be used as the "default" value the first time the script is loaded into an NSC object. Parameters may not be defined using values from previously defined declarations or computations. Only direct numerical values are allowed. The parameter name will appear in the NSC Editor and the value may be changed as any other parameter value for any object.

Constant

CONSTANT name, value
The CONSTANT command defines the name and numerical value of a constant which can subsequently be used in the place of any numerical value.

Object Creation Commands

Object commands create new objects from a list of parameters, and possibly references to existing objects.

Cube

CUBE name, xwidth, ywidth, zwidth
Creates a rectangular volume with the specified width in each direction. The origin is at the center of the cube.

Cone

CONE name, r1, r2, L
Creates a cylindrical or cone shaped volume with the front radial size r1, rear radial size r2, and a length L. Either r1 or r2 may be zero, but not both. L may be positive or negative, but not zero. The origin is at the center of face 1 which lies in the XY plane; the length is along the +Z axis.

Cylinder

CYL name, radius, length
Creates a cylinder with the specified radius and length. Length may be positive or negative, but not zero. The origin is at the center of the face which lies in the XY plane; the length is along the Z axis.

Compound Parabolic Concentrator

CPC name, aperture, angle, length, index
Creates a rotationally symmetric compound parabolic concentrator (CPC). The angle is in degrees. If an index greater than 1.0 is provided, then the angle is the refracted angle in the medium of that index.

Rectangular Compound Parabolic Concentrator

CPCR name, aperture_y, angle_y, length, aperture_x, angle_x, index
Creates a rectangular compound parabolic concentrator (CPCR). The angle is in degrees. If an index greater than 1.0 is provided, then the angle is the refracted angle in the medium of that index.

Elliptical Volume 1

ELLIPSE name, rx, ry, L
Creates a volume with elliptical front and back faces with the specified radial size along the X and Y directions, extruded a length L along the Z direction. L may be positive or negative, but not zero. The origin is at the center of the front face.

Elliptical Volume 2

ELLIPTICAL name, front_x, front_y, back_x, back_y, length
Creates a volume with elliptical front and back faces with the specified radial size along the X and Y directions, extruded a length L along the Z direction. L may be positive or negative, but not zero. The origin is at the center of the front face.

Lens

LENS name, r1, k1, radialAperture1, thick, r2, k2, radialAperture2, bevel1, bevel2
Creates a standard lens. The r1 and r2 values are the front and back radii of curvature (use 0 for flat). The k1 and k2 values are the conic constants. The radial clear aperture of the front and back are radialAperture1 and radialAperture2, respectively, and the center to center thickness is given by thick. The origin is at the center of the front face. Optionally the bevel1 and bevel2 values may be added to the radial edge aperture front and back faces.

Polygon

POLYGON name, n, size, depth
Creates an extruded n-sided polygon. The value n must be at least 3. Size is the distance from the center axis to any vertex on the polygon. The front face lies in the XY plane and the depth is the extruded distance along Z. The depth may be positive or negative, but not zero. One vertex of the polygon lies on the +x axis.

Pyramid

PYRAMID name, n, size, depth
Creates an extruded n-sided pyramid. The value n must be at least 3. Size is the distance from the center axis to any vertex on the polygon. The base face lies in the XY plane and the depth is the extruded distance along Z. The depth may be positive or negative, but not zero. One vertex of the polygon lies on the +x axis.

Slot

SLOT name, radius, spacing, depth
Creates a slot with semi-circular sides, a center to center spacing, and a depth in the Z direction. The origin is at the center of the slot.

Sphere

SPHERE name, radius
Creates a sphere with the specified radius. The origin is the center of the sphere.

Spiral

SPIRAL name, radius, R1, R2, turns, pitch
Creates a spiral shape. The spiral axis is along the +Z axis. The spiral "wire" has a circular cross section of radial size "radius". The radius of the windings from the Z axis to the center of the wire is given by R1 at the beginning of the spiral

and by R2 at the end of the spiral. The turns value is the number of full windings, and need not be an integer. The pitch value is the change in Z coordinate per turn. The radius parameter should be no larger than one third of the smaller of R1 and R2.

Torus

TORUS name, R, r, start_angle, stop_angle
Creates a circular torus volume. A circle of radius r is placed in the YZ plane a distance R along the +Z axis. The angle 0 corresponds to this position of the circle. The circle is spun about the +Y axis from the start angle tothe stop angle to form the torus. The start and stop angles are measured in degrees and must be between -360 and +360.

Sketch Object Commands

Sketch Object commands can be classified under the object creation commands. Sketch Object commands require a user defined sketch (made in Sketch Mode) associated with each sketch Object command.

Prism/Extrusion
EXTRUDE name, sketch, draft_angle, length
Creates a volume using the sketch specified in sketchID. The volume is aligned along the z-axis and forms a prism of the given length. The user may also specify a draft angle through the value d_angle. If no draft angle is present, the value is set to 0.0. The length of the prism cannot be 0. The sketch used in the operation must be closed, and curves within the sketch may not overlap.

Pyramid
PYRAMID_S name, sketchID, length
Creates a pyramidal volume using the sketch specified in sketchID. The volume is aligned along the z-axis and forms a point on the z-axis at the specified length. The length of the pyramid cannot be 0. The sketch used in the operation must be closed, and curves within the sketch may not overlap. The sketch may only use straight line segments.

Revolve
REVOLVE name, sketchID, start_angle, end_angle
Creates a volume made from the revolution of the sketch specified with sketchID. The sketch is roatated from start_angle to end_angle about the y-axis. The sketch may only on either side of the y-axis, and must be a closed shape. The start_angle and end_angle must be between 0 and 360.

Operations Group

Shapes

Array

ARRAY name, object, nx, ny, nz, dx, dy, dz

Creates an array of the geometry defined by object. The geometry defined by the named object is replicated nx, ny, and nz times with a spacing of dx, dy, and dz. The array is always centered in the coordinate system, whether the n value are even or odd. Note this command creates a new object from object1 and the original object is unaltered. Arrays cannot be created if they consist of over 200 individual objects; This means nx * ny * nz may not exceed 200.

Copy

COPY name, object
Creates a new object by making a copy of an existing object. The original object is unaltered.

Ring

RING name, object, x, y, z, cx, cy, cz, count
Creates a ring array of the geometry defined by object. The ring is centered at the point (x, y, z) and the axis of rotation of the part is given by the direction cosines (cx, cy, cz). The total number of replicated objects is given by count. Before using the RING command, the reference object must be offset from the axis of rotation via a MOVE or POSITION command. Note this command creates a new object from object1 and the original object is unaltered. The parameter "count" may not exceed 200 objects.

Chamfer

CHAMFER object, width
Adds a chamfer of the specified width to all sharp edges on the object.

Fillet

FILLET object, radius
Adds a rounded fillet of the specified radius to all sharp edges on the object.

Free

FREE object
Releases the memory associated with the specified object. This command saves system resources, but is not strictly required as objects are automatically freed at the end of the script.

Scale

SCALE object, sx, sy, sz
Scales the named object by the specified factors in the X, Y, and Z directions. The scale values must not be zero, but may be positive or negative. If negative, the object is reversed in the corresponding direction.

Color

COLOR object, R,G,B
Colors an object created by an object creation command with the RGB values specified. Each value must be between 0-255.

Translate

Move

MOVE object, dx, dy, dz
Moves the named object by the specified translation in x, y, and z directions.

Rotate

ROTATE object, angle, cx, cy, cz

Rotates the named object by the specified angle in degrees about the arbitrary axis defined by the x, y, and z direction cosines.

Mirror

MIRROR object, x, y, z, cx, cy, cz
Mirror copies the named object, and reflects the copy relative to a plane defined by the specified (x, y, z) point and normal vector with direction cosines (cx, cy, cz). The original and the copy are then merged to be a single object, whether or not they touch. See also REFLECT, which reflects an object without making a copy.

Reflect

REFLECT object, x, y, z, cx, cy, cz
Reflects the named object relative to a plane defined by the specified (x, y, z) point and normal vector with direction cosines (cx, cy, cz). See also MIRROR, which reflects an object and merges it with the unreflected original object.

Position

POSITION object, x, y, z, cx, cy, cz
Moves and rotates the object to a new coordinate system centered on the specified (x, y, z) point with the local +Z axis pointing along the direction cosines (cx, cy, cz). The original origin is assumed to lie on the new ZY plane.

Boolean

Union

UNION name, object1, object2
Creates a new object that is the Boolean union of object1 and object2.

Difference
a
DIFFERENCE name, object1, object2
Create a new object that is the Boolean difference between object1 and object2. This command is used to remove from object1 the volume represented by object2.

Overlap

OVERLAP name, object1, object2
Create a new object that is the Boolean overlap (logical AND) of object1 and object2. This command is used to select only the volume that is shared by both object1 and object2.

Ray

RAY data, x, y, z, cx, cy, cz, object
The RAY command defines a ray at the specified x, y, and z point, with the specified cx,cy,and cz direction cosines, and then traces the ray to the named object. If the ray intercepts the object, the point of interception is stored in the named data value. The named data value contains three values: the x, y, and z coordinate of the intercept point. These values may be used as constants in subsequent script commands.

Example

Placing a small sphere on the top of a cube

This script defines a cube, then a ray starting above the cube intercepts the top face of the cube, and defines a point where a sphere is placed.

```
CUBE box, 1, 1, 1
RAY point, 0, 10, 0, 0, -1, 0, box
SPHERE ball, 0.3
MOVE ball, point.X, point.Y, point.Z
UNION final, box, ball
```

Note the individual coordinates of the point of intersection

Math Syntax

The Zemax Part Designer is also capable of performing simple mathematical calculations as a substitute for data values within a script. The following operations may be used:

+, -: Addition and subtraction

*, /: Multiplication and division

(): All operations and values may be wrapped within parenthesis

SQRT data_value: the Square root command evaluates the square root of the data value following the command. The input cannot be less than or equal to 0.

SINE, COSI, TANG, ATAN, ASIN, ACOS: The Sine, Cosine, Tangent, Arc Tangent, Arc Sine, and Arc Cosine commands, respectively. These commands are followed by a data value given in radians.

The PARAMETER command is not compatible with these operations.

Tutorial 1: 7-Cell Cluster Concentrator Optic

The following design is meant to house 7 LED light sources and acts to generate an intense beam of light for illuminating small objects. A finished sample file may be found in the C:\[…]\Documents\Zemax\Objects\Part Designer Objects folder with the file name "Cluster_Concentrator.ZPO"

The 7 cell cluster concentrator optic is molded from polycarbonate and sits atop a 7 cell LED array

The concentrator may look like a fairly complex part. Upon closer inspection, the part may be broken down into 3 segments. The profile of the part suggests that it is cut from a single lens part. The large hole in the center of the part is also cut from a lens shape. The radial "flower" shape of the part consists of 6 hexagons, intersected with 6 compound parabolic concentrators (CPC) with lens shaped holes cut at the base of each CPC.

This tutorial will break down the 7-Cell Cluster Concentrator Optic into several simple pieces to create a fully parametric part.

Step 1: Design

Before starting a part, it is important to know which features of a part need to be parameterized. For this tutorial, the overall thickness and diameter of the part will be parameterized. Other measurements representing the diameter of certain gaps and cuts will be represented by the constant, diamRatio. This ratio is attained from the initial specifications of the part. Each of these values is defined below:

```
PARAMETER diameter, 20
PARAMETER thickness, 16
CONSTANT diamRatio, 14.9/38.5
```


Step 2: The Main Lens

The main body of the concentrator is created with the following commands:
```
LENS lensMain, 0, 0.0, diameter, thickness, -40, 0.0, diameter, 0.0, 0.0
```
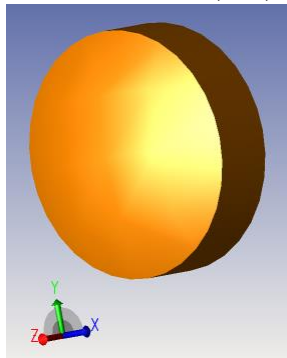


```
LENS frontCut,-
diameter,0.0,diameter*diamRatio,thickness*(2/3),0,0.0,diameter*diamRatio,0.0,0.0
```
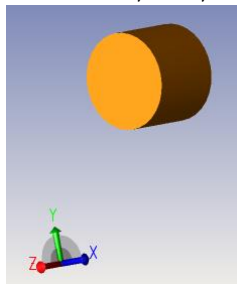


```
RAY frontCutPos, 0,0,diameter*2, 0,0,-1,lensMain
MOVE frontCut, 0,0,frontCutPos.Z*0.8
```
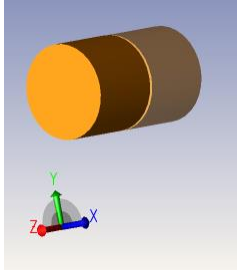
```
DIFFERENCE frontLens, lensMain, frontCut
```
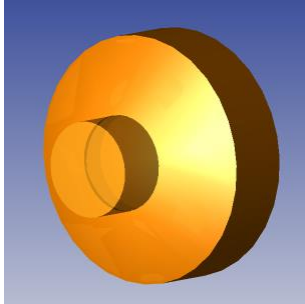

When creating a lens, it is important that each of the parameters abide by the following definition:

$$z = \frac{cr^2}{1 \pm \sqrt{1 - (1 + k)c^2r^2}}$$

Where c is the curvature (the reciprocal of the radius), k is the conic constant and r is half the diameter of the aperture. Values chosen with the LENS command apply to both the front and back surfaces of the lens.

After creating the initial lens, another smaller lens is created to act as the area subtracted from the main lens. The RAY command becomes very effective in positioning parts with respect to other parts as shown in Fig. XVI. The RAY command will grant the user access to the location of a ray intersection specified in the RAY command. The resulting values act similarly to variables created by the CONSTANT command, and can undergo mathematical operations. In this example, the smaller lens needs to be placed close to the front surface (along the +z-axis) of the larger lens. Due to the curvature of the lens, the placement of the smaller lens is not apparent without some calculations. Firing a ray from some arbitrary distance along the z-axis towards the origin will solve this problem. The ray intersection along the lens' surface is stored within the script and can be accessed from another command. The z-coordinate attained by the RAY command indicates a point on the surface of the larger lens, however, the smaller lens does not simply lay tangent to the lens when performing a DIFFERENCE command. This means that a scale factor (simply multiplying the placement by a scale factor by an arbitrary or user defined value) can be used to alter the placement and achieve the desired Boolean result.

Step 3: Base Polygon Shape

The following commands are used to create the base polygon shape. This shape is made up of 6 hexagonal polygons arranged like a honey comb:

```
POLYGON poly1,6,diameter*diamRatio,diameter
```
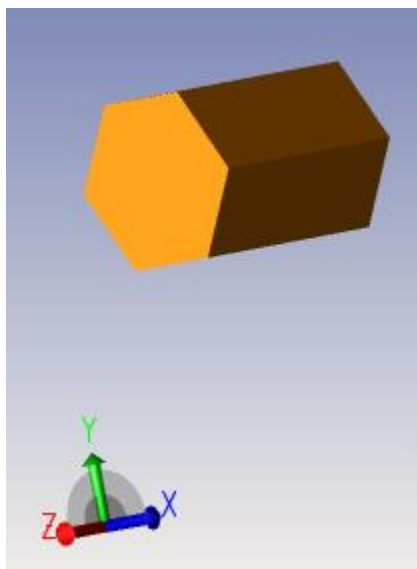
```
COPY poly2, poly1
RAY poly2Pos, 0,diameter,0, 0,-1, 0, poly1
MOVE poly2, 0,2*poly2Pos.Y, 0
```
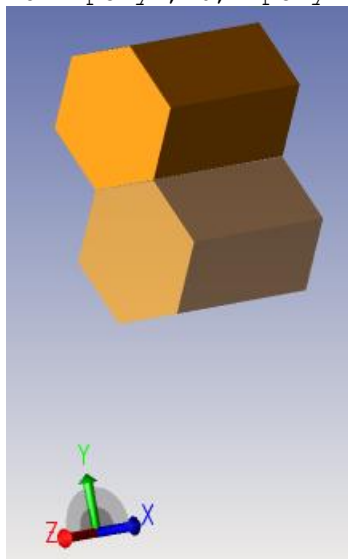


```
RING poly2Ring,poly2,0,0,0,0,0,1,6
```
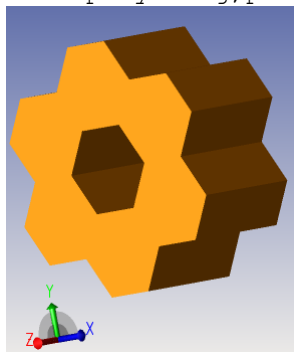
```
UNION polys, poly1,poly2Ring
```
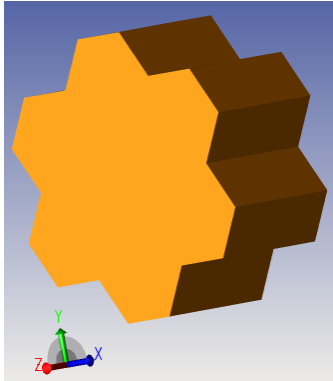

The parameters defining the base polygon utilize the user defined parameters. This ensures that the entire part may be fully parametric within Zemax. The "honey comb" pattern of the polygons is created through a ring array of the initial base polygon about the z-axis. It is important to note that the base shape used in a RING command must first be offset from the specified axis of rotation. Because the geometries are rotated about the origin (axis of rotation will be the z-axis), a copy of the initial polygon must be created and placed on top of the base shape. Once again, the top of the polygon is not clear without some tedious math, and a RAY command is used to find the offset distance of the copied polygon.

It is important to remember that the MOVE and POSITION commands move the origin of a shape. In this example, the origin of the base polygon lays at the origin, at the center of the back face of the polygon. Therefore, the command:

```
MOVE poly2, 0,2*poly2Pos.Y, 0
```

Offsets the origin 2*poly2Pos.Y units along the Y-axis. Use the Object Creation Commands section of this manual to determine the origin of a given geometry.

The initial base polygon and the ring copies are then joined via the UNION command.

The next step requires operations utilized within all prior sections of the tutorial. The following commands detail an array of CPC objects that will be placed behind the newly created polygons.
```
CPC cpc, diameter/5, 5, diameter, 1.0
```
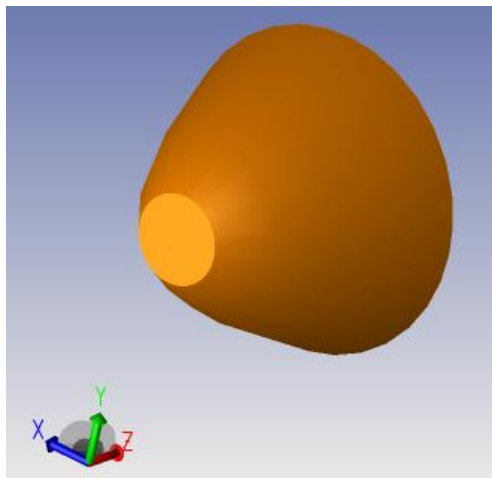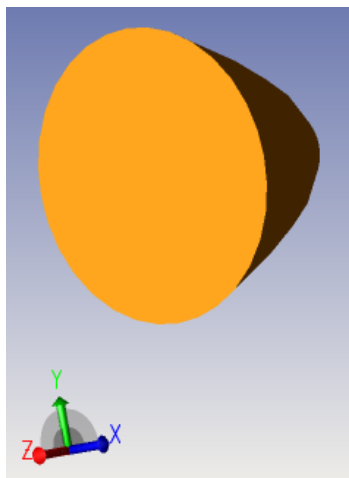
```
LENS cpcCut,0,0,diameter/5,diameter/5,diameter/4,0,diameter/5,0,0
```
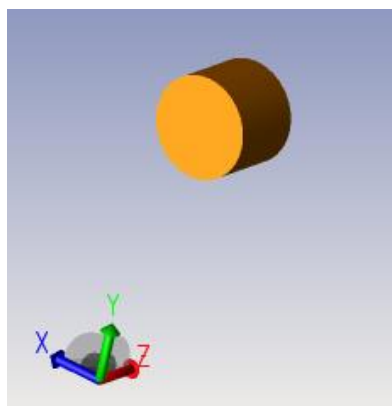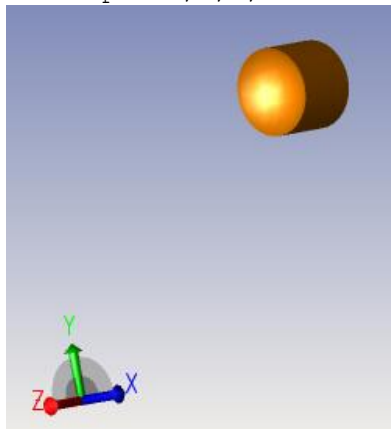


```
DIFFERENCE cpcLobe, cpc, cpcCut
```
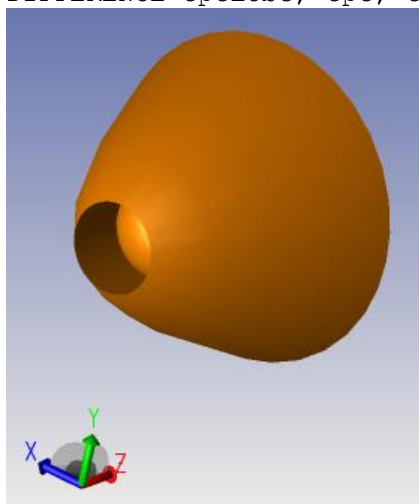


```
COPY cpcLobe2, cpcLobe
```

```
MOVE cpcLobe2, 0, 2*poly2Pos.Y,0
```
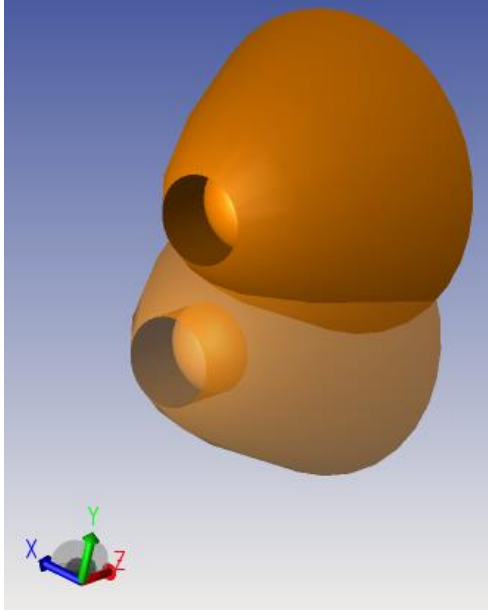
```
RING cpcLobeRing, cpcLobe2, 0,0,0,0,0,1,6
```
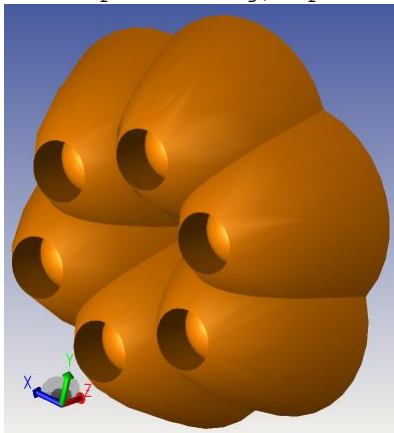


```
UNION cpcLobes, cpcLobeRing, cpcLobe
```
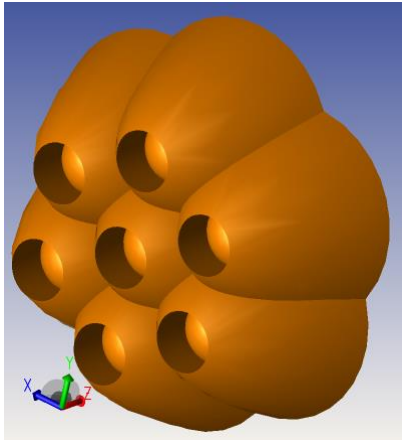

A ring array of 7 compound parabolic concentrators has small air lenses drilled at the rear apertures. Once again, the CPC geometries must be defined by the parameters declared at the beginning of the script. An individual CPC object

---

with a hole is copied and offset along the y-axis using the same values attained by the previous RAY operation. The copied CPC is then rotated using the RING command and combined with the original CPC located at the origin.

Step 5: Finishing the Concentrator

The only Boolean commands utilized thus far have been the DIFFERENCE and UNION commands. The 3rd and final Boolean command is the OVERLAP command. The overlap command takes two objects and creates a new object composed of the intersection between the two objects.
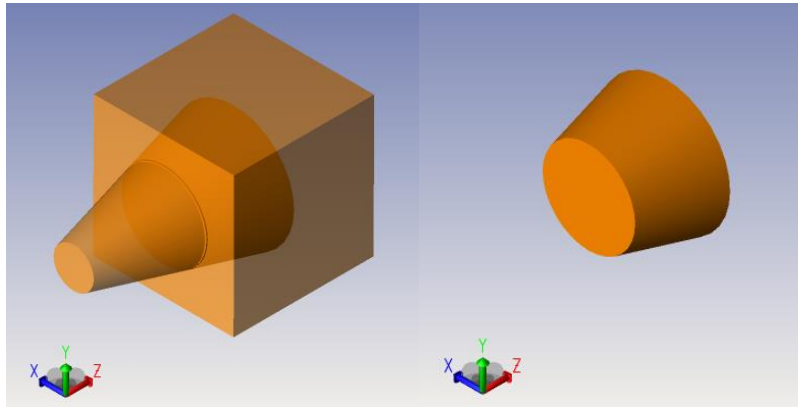
The result of an intersection between a cone and a cube that occurs when using the OVERLAP command.

An OVERLAP with the polygon ring and the CPC ring results in the following shape:

```
OVERLAP baseShape, polys, cpcLobes
```

The images below show the objects that will be overlapped (polys and cpcLobes) as well as the overlapped parts and the resultant object named baseShape.
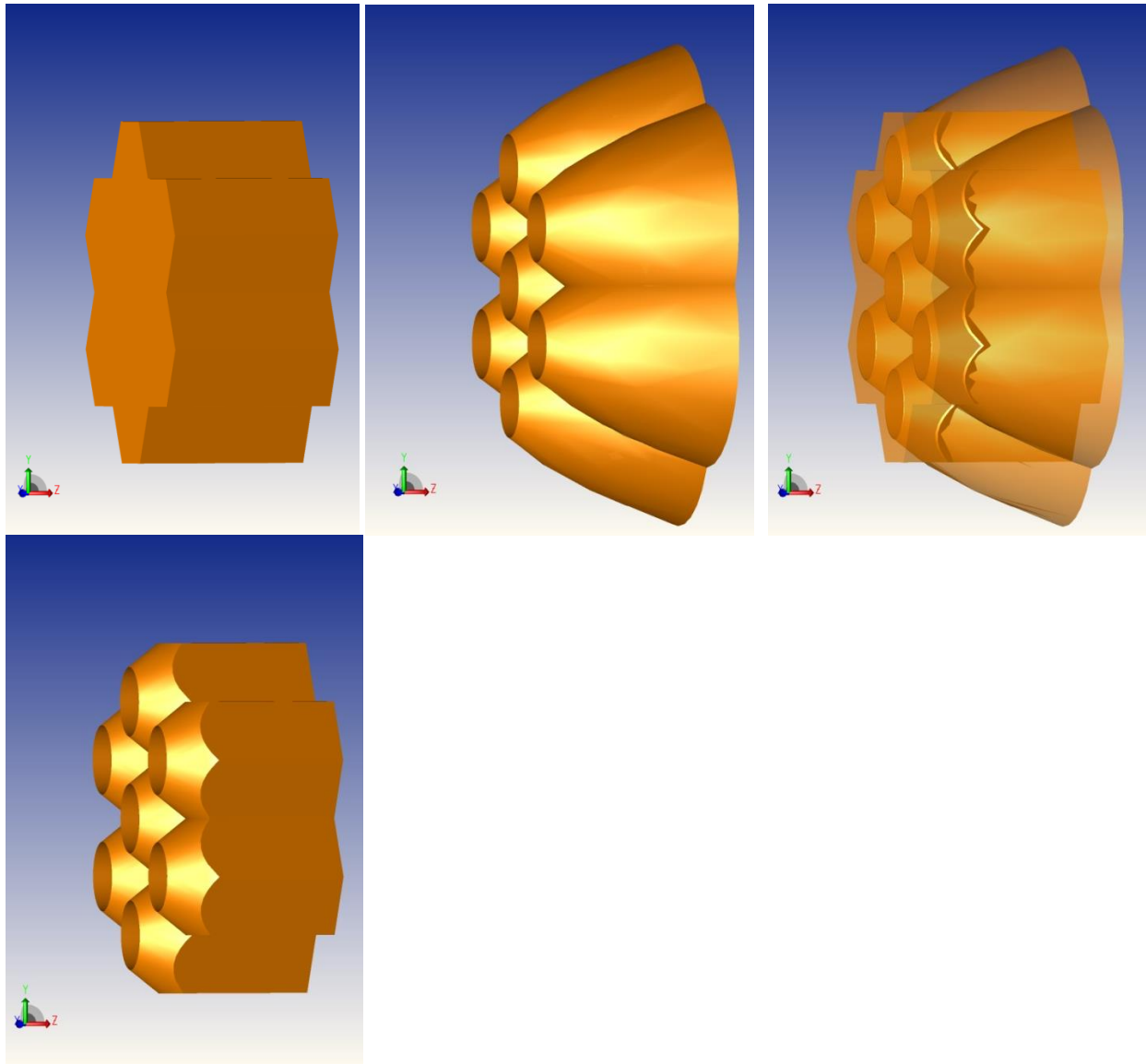
A subsequent OVERLAP command between the new shape and the original large lens results in the object shown below:

```
OVERLAP finalConcentrator, baseShape, frontLens
```
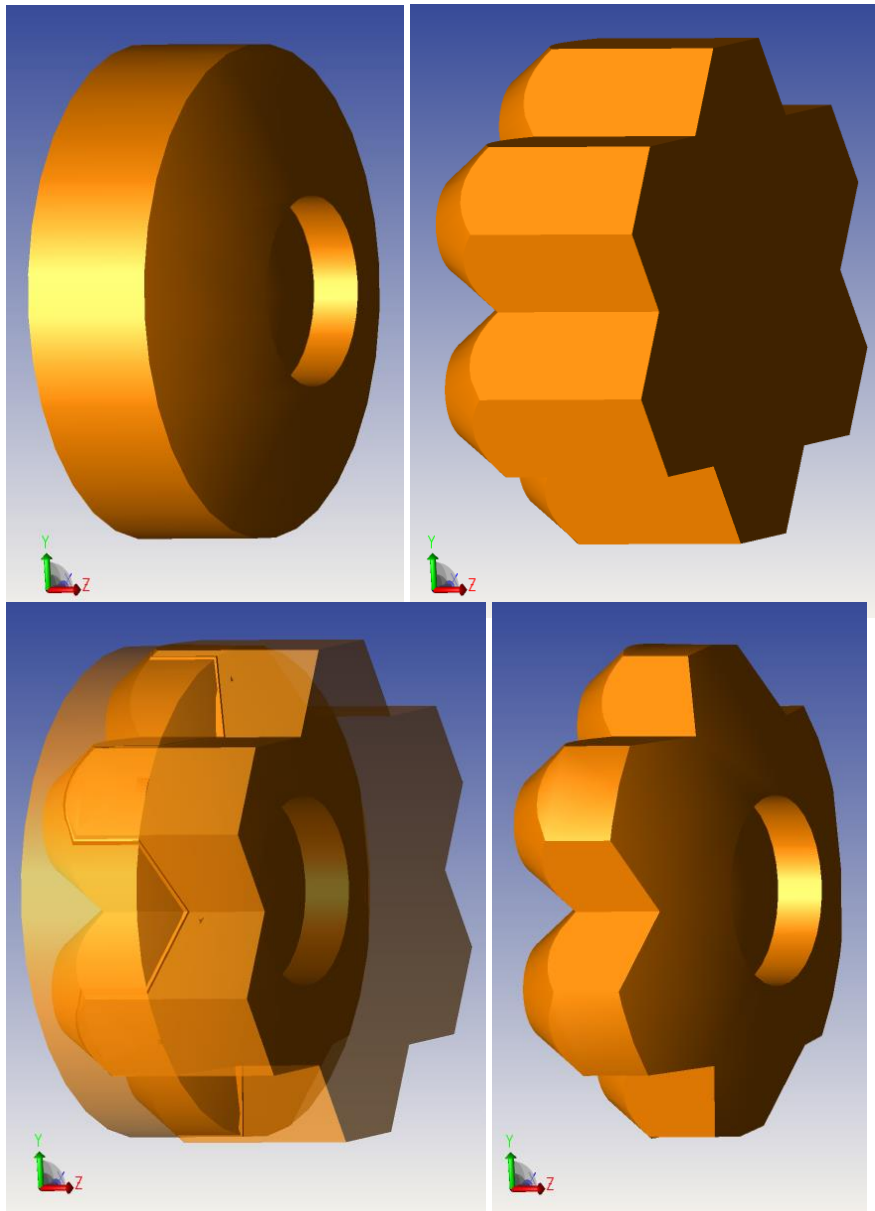
Step 6: Saving the File

The object is now ready to be exported as a CAD file (.IGS, .STEP,.STL, .SAT) or saved as a .ZSO or .ZPO file. It is recommended that parts be saved as .ZPO files, as file features are more robust and can include sketches and gallery data.

To save the file, navigate to the File icon in the System Group and select "save", "save as", or "save gallery". The part may also be exported via the "export as" command

Tutorial 2: Building a Prism from Sketches

In this tutorial, users will utilize the various sketch features within the Zemax Part Designer and learn to optimize a part within the Zemax environment.

User's will create a simple Schmidt-Pechan Roof Prism using the line segment tool as shown below:
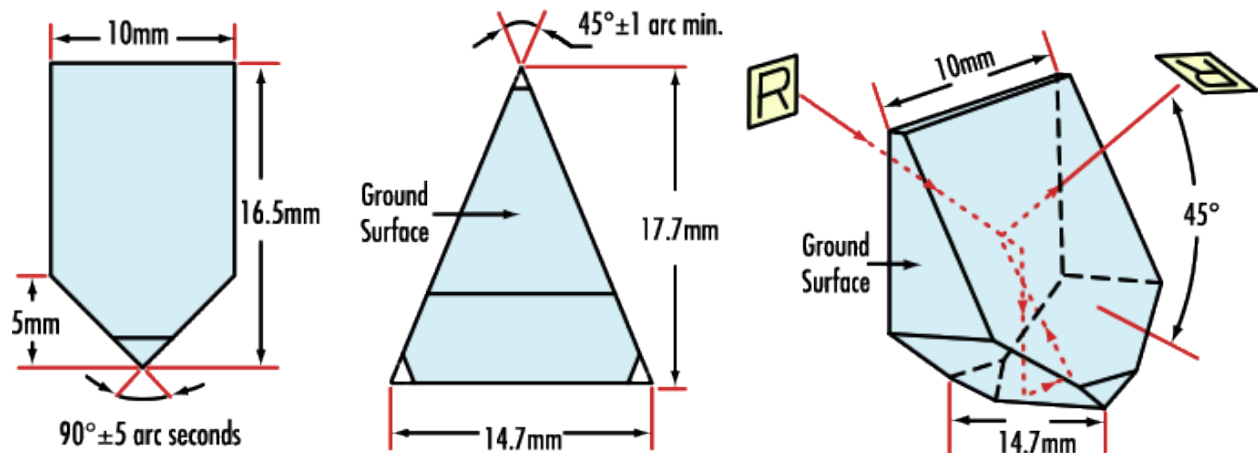
A Schmidt-Pecan Prism and its dimensions.

Step I: Drawing the Pentagonal Base Shape

Begin by opening the Zemax Part Designer and navigating to the sketch mode.
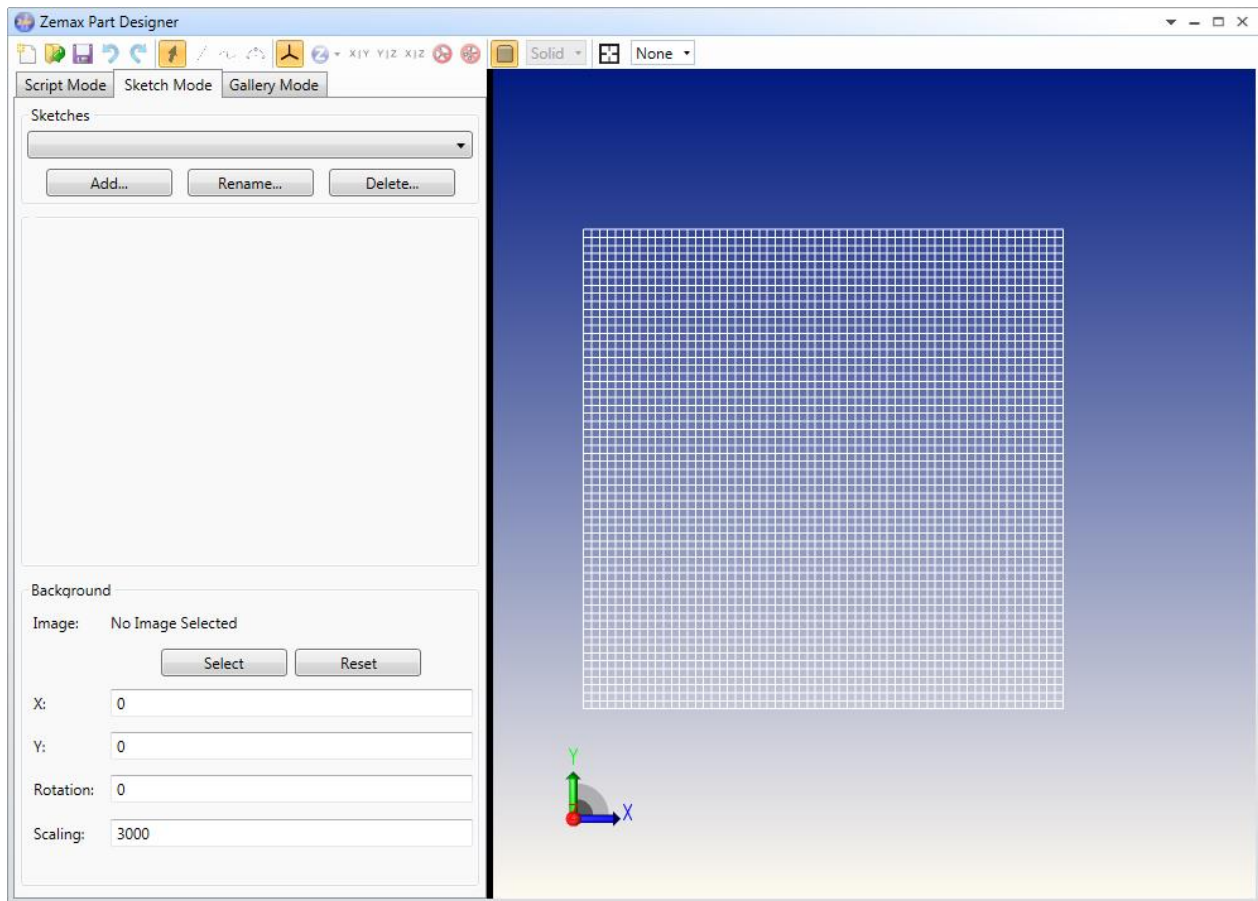
Add a single sketch to the part file by clicking the "Add…" button located below the empty sketch list. This will add an empty sketch entitled "Sketch1" to the list box. Rename this sketch to "pentagon" by clicking the "Rename…" button next to the "Add..." button.
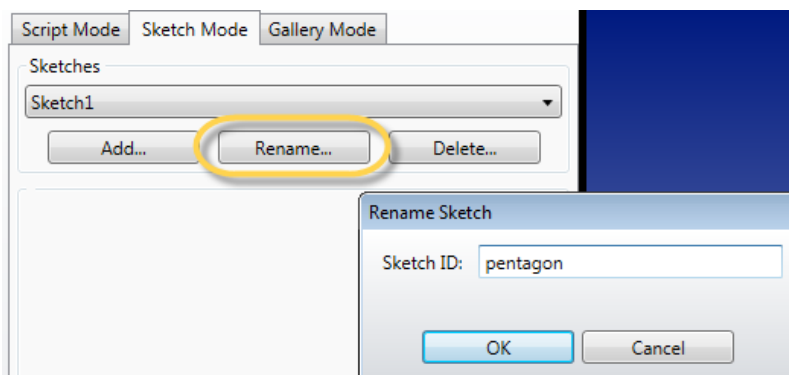
Press "OK" to rename the sketch after entering a valid text string.

Begin the sketch by selecting the Line Tool located along the tool bar at the top of the screen. Plot the following points in the order shown below. You can plot the approximate locations and set values afterword via the Point Control Panel.

Point 1: (-5.000, 10.000)

Point 2: (5.000, 10.000)

Point 3: (5.000, -1.500)

Point 4: (0.000, -6.515)

Point 5: (-5.000, -1.500)

Point 6: (-5.000, 10.000)

Return ZPD into script mode and type the following command into the script editor:

```
EXTRUDE baseShape, pentagon, 0, 20
```

The command instructs ZPD to create an extruded shape, "baseShape", utilizing the sketch, "pentagon", with a draft angle of 0° and a length of 20 along the positive Z-axis (extrusions will always begin at the origin and protrude along the positive Z-axis).
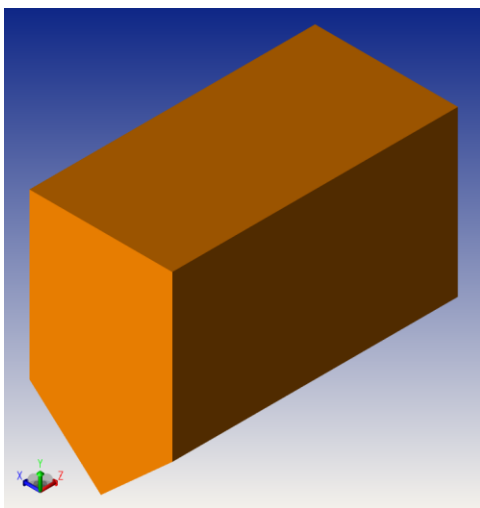
The extruded pentagon extending 20 units out of the origin along the +Z-Axis

Thinking ahead, the final prism must be a triangular cut from the "baseShape" geometry. As all extrusions extend along the +Z-axis, baseShape must be rotated 90° about the y-axis and centered at the origin. Modify the entire script to the following and rebuild:

```
CONSTANT length, 20

EXTRUDE baseShape,pentagon,0,length

MOVE baseShape, 0,0,-(length/2)

ROTATE baseShape, 90, 0, 1,0
```

The variable indicating the length of the extrusion is reprogrammed as a CONSTANT as to be able to modify the length of the extrusion at any time. The variable is not initialized as a PARAMETER because the value used is fairly arbitrary and does not need to be modified within the Zemax Environment.
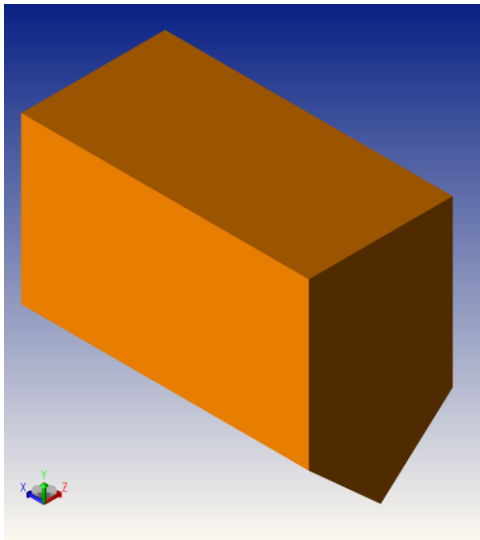
The baseShape is rotated and offset so that the middle of the part is at the origin

Step II: Creating a Triangular Cut

Re-enter Sketch Mode and add a new Sketch titled, "triangle". Using the line tool, plot the following points:

```
Point 1: (0.000, 11.000)

Point 2: (-7.350, -6.515)

Point 3: (7.350, -6.515)
```
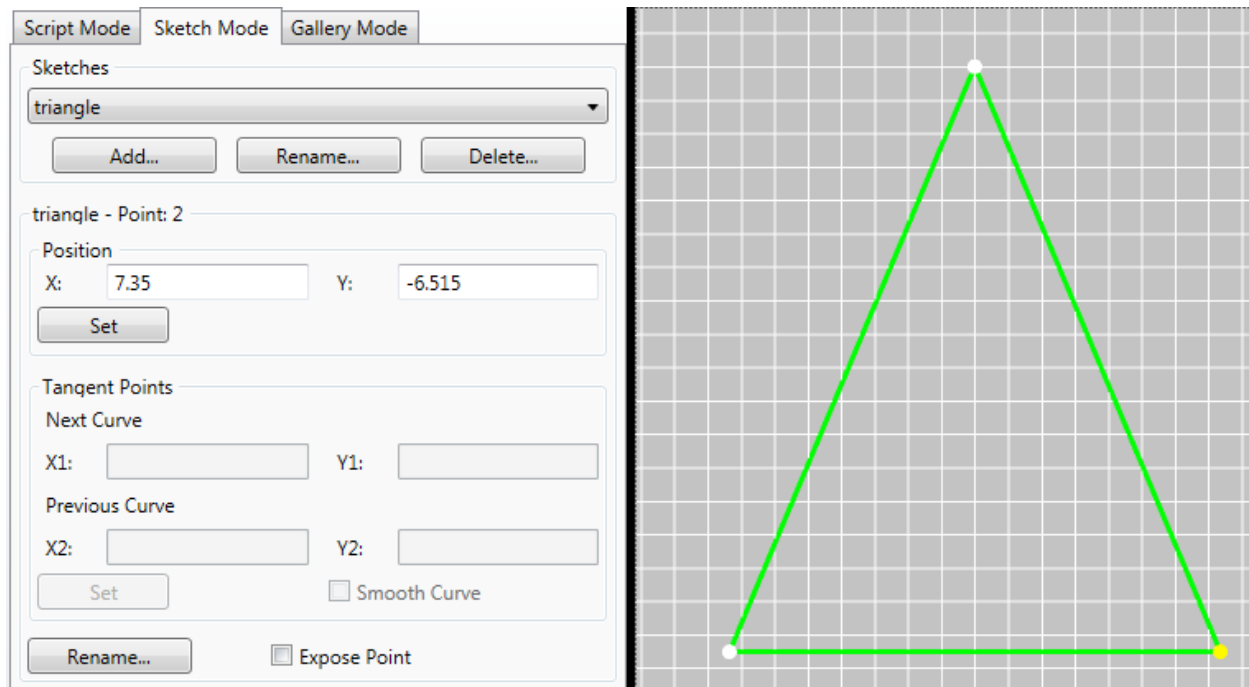
Plotting the control points for the triangle cross section.

Navigate back to Script Mode and enter the following commands:

```
EXTRUDE cut,triangle,0,length
```
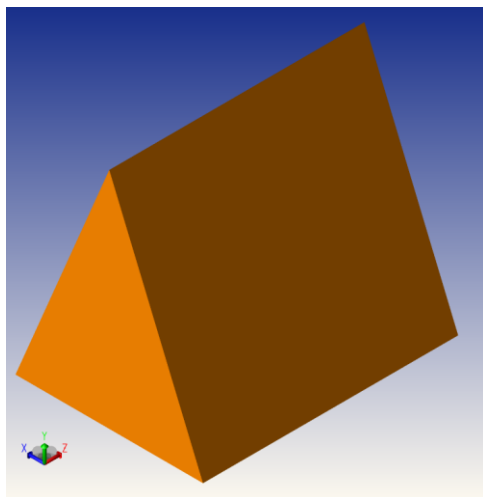


```
MOVE cut, 0,0,-length/2
OVERLAP final, baseShape, cut
```
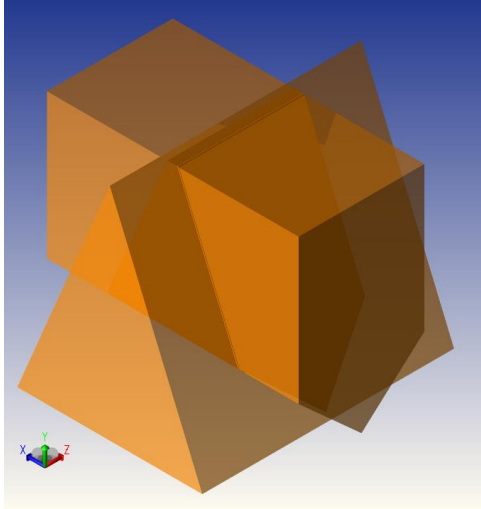
The extruded part is created with the specified length, centered, and cut from the initial baseShape Object.

The Schmidt-Pechan Roof Prism is now complete.

Step III: Optimizing the part within Zemax

Depending on the user's usage of the resulting part, there are several ways in which this part can be optimized. In the case of this particular part, light typically passes from one pentagonal face of the prism to the other. One primary factor in this behavior is the angle adjoining the 2 pentagonal faces.
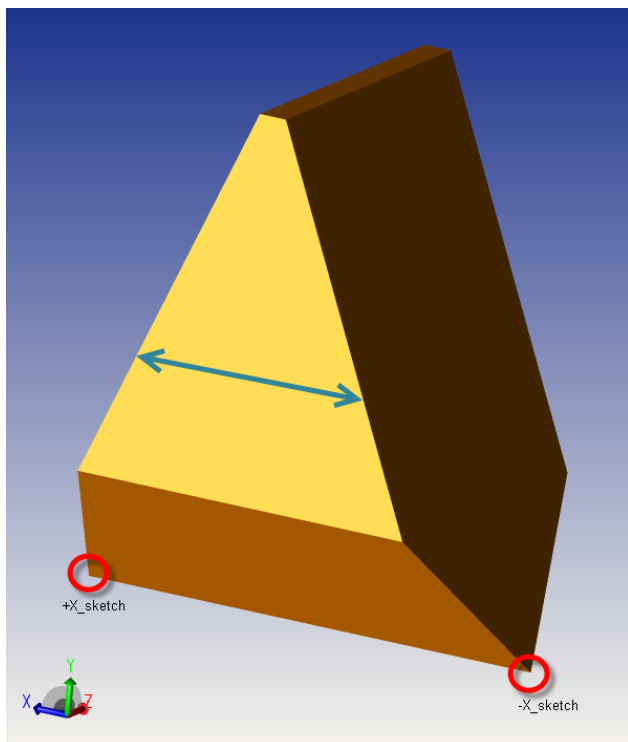
Adjusting the points circled in red will change the angle marked by the blue arrows.

To increase or decrease this angle, Zemax must be able to adjust the two points indicated in Figure GGSDA. Navigate into Sketch Mode and "expose" these two points.
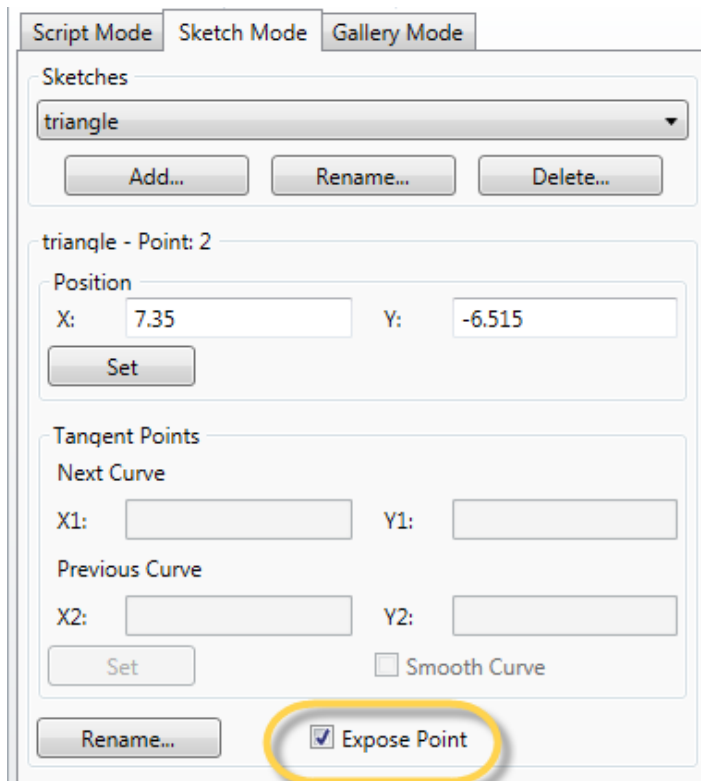
Save the file into the local "My Documents"-> "Zemax"-> "Objects"->"Part Designer Objects" folder. The final part should appear identical to the resulting part from Step II.

Load the part into a Non-Sequnetial Component Editor as a CAD Part: Zemax Part Designer.
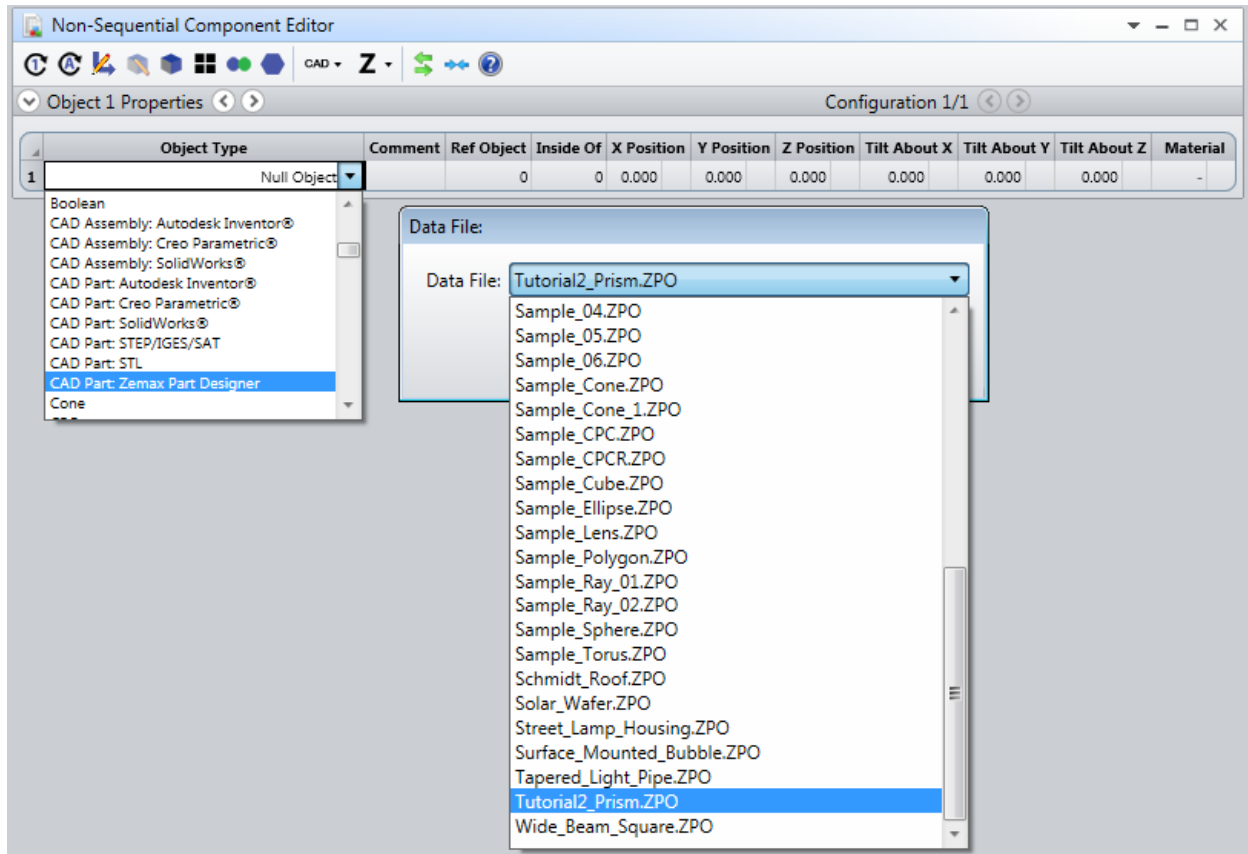
Upon opening the file, the Non-Sequential Editor should expose the x, y coordinates of the 2 selected points. Each value is represented by 1 Parameter within the Editor, so a total of 4 variables should appear.

The exposed points will appear in the non-sequential mode editor as highlighted. In this case, the points were renamed to "+Xsketch" and "-Xsketch"

As the triangular cross section of the prism is symmetrical, add a pick up solve to parameter 3 (-Xsketch.x), scaling the parameter to be of the opposite sign from "+Xsketch.x" at all times. Adjusting parameter 2 will now result in an equal and opposite change for Parameter 3.

+Xsketch.x	+Xsketch.y	-Xsketch.x		-Xsketch.y
7.350	-6.515	-7.350	P	-6.515

Parameter 3 solve on object 1

Solve Type:	Pickup
From Object::	1
Scale Factor:	-1
Offset:	0
From Column	Parameter 1
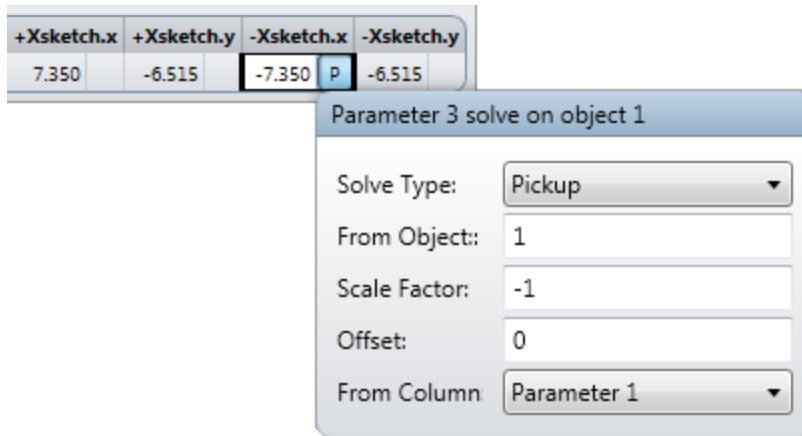
Add a pickup solve to parameter 3 to ensure that Zemax will adjust the points symmetrically.

The y coordinates of these control points can be ignored as they will remain fixed. Finally, Parameter 1 (+Xsketch.x) must now be made into a variable to be used for optimization.

Parameter 1 can now be used as a variable within various merit function parameters and optimization.
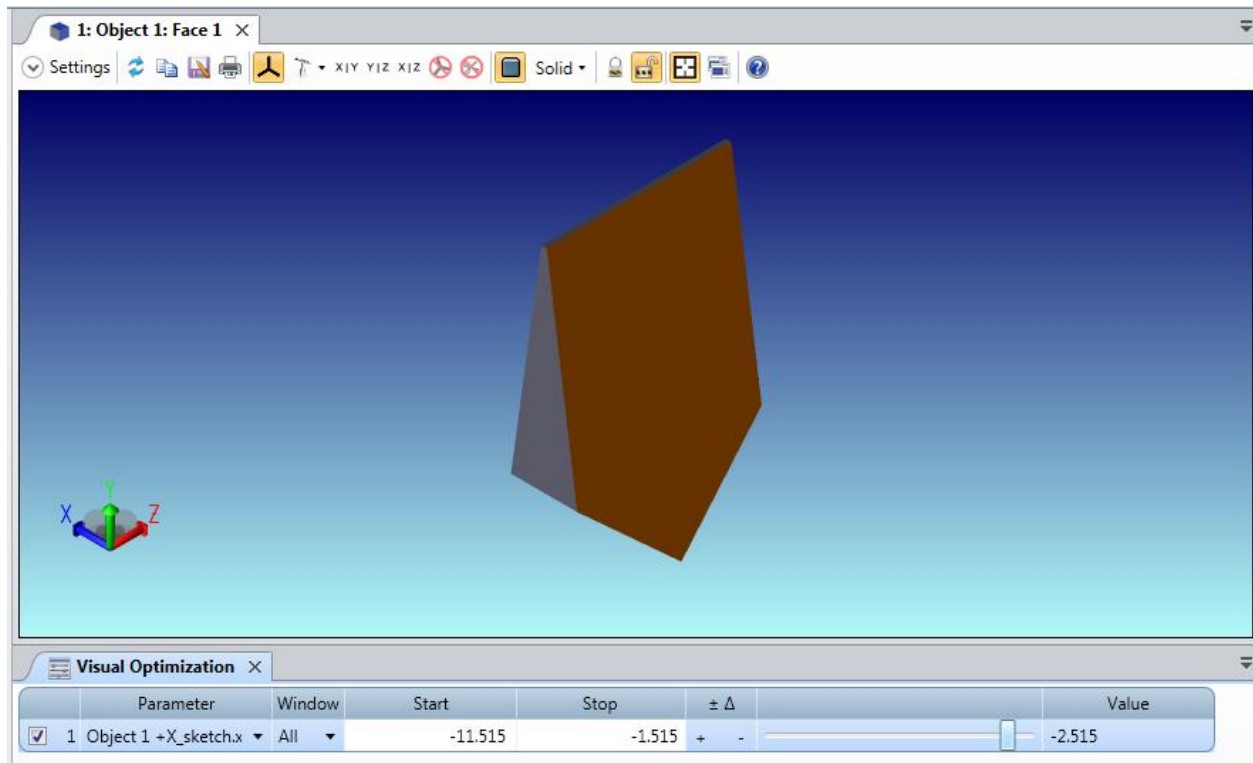
Using the Visual Optimizer slider to adjust the shape of the Prism

To increase the stability of the part, users need to add Merit Function parameters to limit the range of values a point can take. This ensures that only physical and valid parts are created.

Keyboard Shortcuts

These are the default keyboard shortcuts in Sequential and Non-Sequential UI Mode.

To customize the shortcut keys, see the Shortcut Keys section of the Project Preferences.

Sequential Mode

Ctrl + **A**   Next Configuration

Ctrl + **B**   Paraxial Gaussian Beam Data Analysis

Ctrl + **C**   Copy

Ctrl + **E**   Zemax Element Drawing

Ctrl + **F**   Opens Field Data

Ctrl + **G**   Expands and Minimizes System Explorer sidebar (aka General System dialog box)

Ctrl + **H**   Prescription Data

Ctrl + **I**   Interferogram

Ctrl + **J**   Geometric Image Analysis

Ctrl + **K**   Wireframe Layout

Ctrl + **L**   Cross Section aka 2D Layout

Ctrl + **M**   FFT MTF

Ctrl + **N**   New Lens File

Ctrl + **O**   Open

Ctrl + **P**   Print

Ctrl + **Q**   Quit

Ctrl + **R**   Ray Fan

Ctrl + **S**   Save

Ctrl + **T**   Tolerancing

Ctrl + **U**   Updates Current Window

Ctrl + **V**   Paste

Ctrl + **W**   Wavelength Data

Ctrl + **X**	Cut
Ctrl + **Y**	Ray Trace
Ctrl + **Z**	Make Variable

Ctrl + **Shift** + **A**	Last Configuration
Ctrl + **Shift** + **B**	Reverse Elements
Ctrl + **Shift** + **D**	NSC Group Editor
Ctrl + **Shift** + **E**	ISO Element Drawing
Ctrl + **Shift** + **F**	Add Fold Mirror
Ctrl + **Shift** + **G**	Global Search aka Global Optimization
Ctrl + **Shift** + **H**	Hammer Optimization
Ctrl + **Shift** + **J**	Partially Coherent Image Analysis
Ctrl + **Shift** + **K**	Solid Model (Obsolete)
Ctrl + **Shift** + **L**	3D Layout
Ctrl + **Shift** + **N**	Geometric Encircled Energy aka Geometric Enclosed Energy
Ctrl + **Shift** + **O**	Local Optimization
Ctrl + **Shift** + **P**	Dispersion Diagram
Ctrl + **Shift** + **Q**	Quick Focus
Ctrl + **Shift** + **R**	OPD Ray Fan
Ctrl + **Shift** + **S**	Spot Diagram
Ctrl + **Shift** + **U**	Update All
Ctrl + **Shift** + **V**	Visual Optimizer
Ctrl + **Shift** + **X**	System Data
Ctrl + **Shift** + **Z**	Zernike Fringe Coefficients

F1	Help
F3	Undo
Ctrl + **F3**	Redo
F4	Materials Catalog aka Glass Catalogs
F5	Lens Catalogs
F6	Merit Function Editor
F7	Multi-Configuration Editor
F8	Formerly Extra Data Editor; now obsolete
F9	Edit/Run ZPL Macros
F10	About

Shift + **F1** Lens Data Editor

Shift + **F2** Tolerance Data Editor

Shift + **F3** Non-Sequential Component Editor

Ctrl + **Tab** Next Window

Ctrl + **Shift** + **Tab** Last Window

Non-Sequential UI Mode

Ctrl + **A** Next Configuration

Ctrl + **C** Copy

Ctrl + **D** NSC Ray Trace

Ctrl + **G** Expands and Minimizes System Explorer sidebar (aka General System dialog box)

Ctrl + **H** Prescription Data

Ctrl + **L** 3D Layout

Ctrl + **N** New Lens File

Ctrl + **O** Open

Ctrl + **P** Print

Ctrl + **Q** Quit

Ctrl + **S** Save

Ctrl + **T** Tolerancing

Ctrl + **U** Updates Current Window

Ctrl + **V** Paste

Ctrl + **W** Wavelength Data

Ctrl + **X** Cut

Ctrl + **Z** Make Variable

F1 Help

F3 Undo

Ctrl + **F3** Redo

F4 Materials Catalog

F5 Lens Catalogs

F6 Merit Function Editor

F7 Multi-Configuration Editor

F9 Edit/Run ZPL Macros

F10 About

Editors

Ctrl +Left   Move cursor 5 columns left

Ctrl +Right   Move cursor 5 columns right

Ctrl +Up   Move cursor to first row

Ctrl +Down   Move cursor to last row

Tab   Move cursor 1 column right

Shift + Tab   Move cursor 1 column left

Home   Move cursor to first row, first column cell

End   Move cursor to last row, first column cell

Page Up   Move cursor to first row, current column cell

Page Down   Move cursor to last row, current column cell

Graphics and Analysis Windows

Home   Zoom in

End   Zoom out

Page Up   Rotate layout

Page Down   Rotate layout

Arrows   Rotate layouts OR toggle surface number

Ctrl + Tab   Next Window

Conventions and Definitions

This chapter describes the conventions and defines terminology used throughout this manual. Most of the conventions and terms Zemax uses are common in the optics industry, however there may be some important differences.

Active configuration

The active configuration is the configuration currently being displayed in the lens data editor. For details see the Using Multiple Configurations reference file, or the Multiple Configuration Editor help file.

Angular magnification

The ratio of the paraxial image space chief ray angle to the paraxial object space chief ray angle. The angles are measured with respect to the paraxial entrance and exit pupil locations.

Apodization

Apodization refers to the uniformity of illumination in the entrance pupil of the system. By default, the pupil is always illuminated uniformly. However, there are times when the pupil should have a non-uniform illumination. For this purpose, Zemax supports pupil apodization, which is a variation of amplitude over the pupil.

Three types of pupil apodization are supported: uniform, Gaussian, and tangential. For each type (except uniform), an apodization factor determines the rate of variation of amplitude in the pupil. See the discussion on apodization types and factors in the Aperture section of the System Explorer.

Zemax also supports user defined apodizations, which may be placed on any surface. Surface apodizations behave differently than pupil apodizations, because surfaces need not be located at a pupil. For more information on surface apodizations, see the User Defined surface.

Back focal length

Zemax defines the back focal length as the distance along the Z axis from the last surface made of glass to the paraxial image surface for the object at infinite conjugates. If no surfaces are made of glass, the back focal length is the distance from surface 1 to the paraxial image surface for the object at infinite conjugates.

Cardinal planes

The term cardinal planes (sometimes called cardinal points) refers to those special conjugate positions where the object and image surfaces have a specific magnification. The cardinal planes include the principal planes, where the lateral magnification is +1, the anti-principal planes, where the lateral magnification is -1, the nodal planes, where the angular magnification is +1, the anti-nodal planes, where the angular magnification is -1, and the focal planes, where the magnification is 0 for the image space focal plane and infinite for the object space focal plane.

Except for the focal planes, the cardinal planes are conjugates with each other, that is, the image space principal plane is conjugate with the object space principal plane, etc. If the lens has the same index in both object space and image space, the nodal planes are identical to the principal planes.

Zemax lists the distance from the image surface to the various image space planes, and lists the distance from the first surface to the various object space planes.

Chief ray

If there is no vignetting, and there are no aberrations, the chief ray is defined to be the ray that travels from a specific field point, through the center of the entrance pupil, and on to the image surface. Note that without vignetting or aberrations, any ray passing through the center of the entrance pupil will also pass through the center of the stop and the exit pupil.

When vignetting factors are used, the chief ray is then considered to be the ray that passes through the center of the vignetted pupil, which means the chief ray may not necessarily pass through the center of the stop.

If there are pupil aberrations, and there virtually always are, then the chief ray may pass through the center of the paraxial entrance pupil (if ray aiming is off) or the center of the stop (if ray aiming is on), but generally, not both.

If there are vignetting factors which decenter the pupil, then the chief ray will pass through the center of the vignetted entrance pupil (if ray aiming is off) or the vignetted stop surface (if ray aiming is on).

The common convention used is that the chief ray passes through the center of the vignetted pupil, while the principal ray passes through the center of the unvignetted stop. Zemax never uses the principal ray. Most calculations are referenced to the chief ray or the centroid. Note the centroid reference is generally superior because it is based upon the aggregate effect of all the rays that actually illuminate the image surface, and not on the arbitrary selection of one ray which is "special".

Coordinate axes

The optical axis is the Z axis, with the initial direction of propagation from the object being the positive Z direction. Mirrors can subsequently reverse the direction of propagation. The coordinate system is right handed, with the sagittal X axis being oriented "into" the monitor on a standard layout diagram. The tangential Y axis is vertical.

The direction of propagation is initially left-to-right, down the positive Z axis. After an odd number of mirrors the beam physically propagates in a negative Z direction. Therefore, all thicknesses after an odd number of mirrors should be negative.

Diffraction limited

The term diffraction limited implies that the performance of an optical system is limited by the physical effects of diffraction rather than imperfections in either the design or fabrication. A common means of determining if a system is diffraction limited is to compute or measure the optical path difference. If the peak to valley OPD is less than one quarter wave, then the system is said to be diffraction limited.

There are many other ways of determining if a system is diffraction limited, such as Strehl ratio, RMS OPD, standard deviation, maximum slope error, and others. It is possible for a system to be considered diffraction limited by one method and not diffraction limited by another method.

On some Zemax plots, such as the MTF or Diffraction Encircled Energy, the diffraction limited response is optionally shown. This data is usually computed by tracing rays from a reference point in the field of view. Pupil apodization, vignetting, F/#'s, surface apertures, and transmission may be accounted for, but the optical path difference is set to zero regardless of the actual (aberrated) optical path.

For systems which include a field point at 0.0 in both x and y field specifications (such as 0.0 x angle and 0.0 y angle), the reference field position is this axial field point. If no (0, 0) field point is defined, then the field coordinates of field position 1 are used as the reference coordinates instead.

Edge thickness

Zemax defines the edge thickness of a surface as:

$$E_i = Z_{i+1} - Z_i + T_i$$

where is the sag of the surface, is the sag of the next surface, and is the axial thickness of the surface. The sag values are computed at the +y semi-diameter of their respective surfaces; Note that edge thickness is computed for the +y radial aperture, which may be inadequate if the surface is not rotationally symmetric, or if surface apertures have been placed upon either of the surfaces.

Edge thickness solves use a slightly different definition of edge thickness. For edge thickness solves only, the sag of the i+1 surface is computed at the semi-diameter of surface i. This method avoids a possible infinite loop in the calculation, since changing the thickness of surface i may alter the semi-diameter of surface i+1, if the latter semi-diameter is in "automatic" mode and the value is based upon ray tracing. See "Thickness: Edge thickness" and "Entering semi-diameter data" for more information.

Effective focal length

The distance from the rear principal plane to the paraxial image surface. This is calculated for infinite conjugates. Principal plane calculations are always based upon paraxial ray data. The effective focal length is always referenced to an index of refraction of 1.0, even if the image space index is not unity.

Entrance pupil diameter

The diameter in lens units of the paraxial image of the stop in object space.

Entrance pupil position

The paraxial position of the entrance pupil with respect to the first surface in the system. The first surface is always surface 1, not the object surface, which is surface 0.

Exit pupil diameter

The diameter in lens units of the paraxial image of the stop in image space.

Exit pupil position

The paraxial position of the exit pupil with respect to the image surface.

Field angles and heights

Field points may be specified as angles, object heights (for systems with finite conjugates), paraxial image heights, or real image heights. Field angles are always in degrees. The angles are measured with respect to the object space z axis and the paraxial entrance pupil position on the object space z axis. Positive field angles imply positive slope for the ray in that direction, and thus refer to negative coordinates on distant objects. Zemax converts x field angles (α_x) and y field angles (α_y) to ray direction cosines using the following formulas:

$$\tan \alpha_x = \frac{l}{n}$$

$$\tan \alpha_y = \frac{m}{n}$$

$$l^2 + m^2 + n^2 = 1$$

where l, m, and n are the x, y, and z direction cosines.

If object or image heights are used to define the field points, the heights are measured in lens units. When paraxial image heights are used as the field definition, the heights are the paraxial image coordinates of the primary wavelength chief ray on the paraxial image surface, and if the optical system has distortion, then the real chief rays will be at different locations. When real image heights are used as the field definition, the heights are the real ray coordinates of the primary wavelength chief ray on the image surface.

Zemax uses normalized field coordinates for many features. For information on how field coordinates are normalized, see the "Normalized field coordinates" definition. To set the field type and values, see Fields.

Float by stop size

Float by stop size is one of the system aperture types supported by Zemax. This phrase refers to the fact that the entrance pupil position, object space numerical aperture, image space F/#, and stop surface radius all are specified if just one of them is specified. Therefore, setting the stop radius, and then allowing the other values to be whatever they are, is a perfectly valid way of defining the system aperture. It is particularly handy when the stop surface is a real, unchangeable aperture buried in the system, such as when designing null corrector optics.

Ghost reflections

Ghost reflections are spurious, unwanted images formed by the small amount of light which reflects off of, rather than refracts through a lens face. For example, the multiple images of the aperture stop visible in photographs taken with the sun in the field of view are caused by ghost reflections. Ghost images can be problematic in imaging systems and in high power laser systems.

Glasses

Glasses are entered by name in the glass column. Available glasses may be reviewed, and new ones entered using the glass catalog tool. See the Chapter "Using Glass Catalogs" for details.

Blanks are treated as air, with unity index. Mirrors can be specified by entering "MIRROR" for the glass type, although this name will not appear in the glass catalog. The index of refraction of the mirror space is always equal to the index of refraction of the media before the mirror.

For information on the effect of temperature and pressure on index of refraction data, see Defining temperature and pressure.

Hexapolar rings

Zemax usually selects a ray pattern for you when performing common calculations such as spot diagrams. The ray pattern refers to how a set of rays is arranged on the entrance pupil. The hexapolar pattern is a rotationally symmetric means of distributing a set of rays. The hexapolar pattern is described by the number of rings of rays around the central ray. The first ring contains 6 rays, oriented every 60 degrees around the entrance pupil with the first ray starting at 0 degrees (on the x-axis of the pupil). The second ring has 12 rays (for a total of 19, including the center ray in ring "0"). The third ring has 18 rays. Each subsequent ring has 6 more rays than the previous ring.

Many features which require a sampling parameter to be specified (such as the spot diagram) use the number of hexapolar rings as a convenient means of specifying the number of rays. If the hexapolar sampling density is 5, it does not mean that 5 rays will be used. A sampling of 5 means $1 + 6 + 12 + 18 + 24 + 30 = 91$ rays will be used.

Image space F/#

Image space F/# is the ratio of the paraxial effective focal length calculated at infinite conjugates over the paraxial entrance pupil diameter. Note that infinite conjugates are used to define this quantity even if the lens is not used at infinite conjugates.

Image space numerical aperture (NA)

Image space NA is the index of image space times the sine of the angle between the paraxial on-axis chief ray and the paraxial on-axis +y marginal ray calculated at the defined conjugates for the primary wavelength.

Lens units

Lens units are the primary unit of measure for the lens system. Lens units apply to radii, thicknesses, apertures, and other quantities, and may be millimeters, centimeters, inches, or meters.

Marginal ray

The marginal ray is the ray that travels from the center of the object, to the edge of the entrance pupil, and on to the image surface.

If there is vignetting, Zemax extends this definition by defining the marginal ray to be at the edge of the vignetted entrance pupil. If ray aiming is on, then the marginal ray is at the edge of the vignetted stop.

See also the definition of the chief ray.

Maximum field

The maximum field is the minimum radial coordinate that would enclose all the defined field points if the x and y values of each field point were plotted on an Cartesian XY plot. The maximum field is measured in degrees if the field type is angles, or in lens units for object height, paraxial image height, or real image height. The field type is described in the "Field angles and heights" definition. To set the field type and values, see Fields.

Non-paraxial systems

The term non-paraxial system refers to any optical system which cannot be adequately represented by paraxial ray data. This generally includes any system with tilts or decenters, strong aspheres, axicons, holograms, gratings, cubic splines, ABCD matrices, gradient index, diffractive components, or non-sequential surfaces.

A great deal of optical aberration theory has been developed for systems with conventional refractive and reflective components in rotationally symmetric configurations. This includes Seidel aberrations, distortion, Gaussian beam data, and virtually all first order properties such as focal length, F/#, and pupil sizes and locations. All of these values are calculated from paraxial ray data.

If the system being analyzed contains any of the non-paraxial components described, then any data computed based upon paraxial ray tracing cannot be trusted. Zemax will generally use exact real rays rather than paraxial rays for ray tracing through these surfaces and components.

A system which is well described by paraxial optics will have the general property that the real and paraxial marginal ray data converge as the radial entrance pupil coordinate of the rays being traced tends toward zero.

Non-sequential ray tracing

Non-sequential ray tracing means rays are traced only along a physically realizable path until they intercept an object. The ray then refracts, reflects, or is absorbed, depending upon the properties of the object struck. The ray then continues on a new path. In non-sequential ray tracing, rays may strike any group of objects in any order, or may strike the same object repeatedly; depending upon the geometry and properties of the objects.

See also the definition of "Sequential ray tracing".

Normalized field coordinates

Normalized field coordinates are used in both the Zemax program and documentation. There are two normalized field coordinates: Hx and Hy. Normalized field coordinates are convenient because useful field locations may be defined in a manner that does not change with the individual field definitions or field of view of the optical system. For example, the normalized field coordinate (0, 1) is always at the top of the field of view, whether the field points are defined as angles or heights, and regardless of the magnitude of the field coordinates.

There are two methods used to normalize fields: radial and rectangular. The choice of field normalization method is made on the Field Data dialog; for a description see the "Field angles and heights" definition.

Radial field normalization

If the field normalization is radial, then the normalized field coordinates represent points on a unit circle. The radius of this unit circle, called the maximum radial field, is given by the radius of the field point farthest from the origin in field coordinates. The maximum radial field magnitude is then used to scale all fields to normalized field coordinates. Real field coordinates can be determined by multiplying the normalized coordinates, H_x and H_y, by the maximum radial field magnitude:

$$f_x = H_x F_r, \text{ and } f_y = H_y F_r,$$

where F_r is the maximum radial field magnitude and f_x and f_y are the field coordinates in field units.
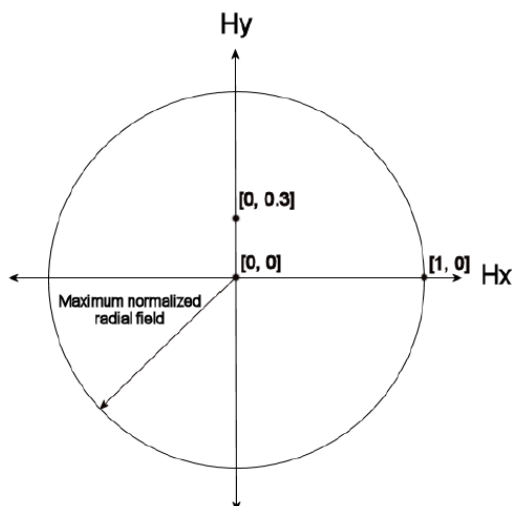
For example, suppose 3 field points are defined in the (x, y) directions using object height in lens units at (0.0, 0.0), (10.0, 0.0), and (0.0, 3.0). The field point with the maximum radial coordinate is the second field point, and the maximum radial field is therefore 10.0. The normalized coordinate (Hx = 0, Hy = 1) would refer to the field coordinates (0.0, 10.0). The normalized coordinate (Hx = 1, Hy = 0) would refer to the field coordinates (10.0, 0.0). Note the normalized field coordinates can define field coordinates that do not correspond to any defined field point. The maximum radial field is always a positive value.

If a fourth field point at (-10.0, -3.0) were added in the above example, the maximum radial field would become

$$F_r = \sqrt{3^2 + 10^2} ,$$

or approximately 10.44031. Normalized coordinates should always be between -1 and 1, and should also meet the condition:

$$H_x^2 + H_y^2 \leq 1 .$$

Otherwise, the field point lies outside of the maximum radial field.

Rectangular field normalization

If the field normalization is rectangular, then the normalized field coordinates represent points on a unit rectangle. The x and y direction widths of this unit rectangle, called the maximum x field and maximum y field, are defined by the largest absolute magnitudes of all the x and y field coordinates. The maximum x and y field magnitudes are then used to scale all fields to normalized field coordinates. Real field coordinates can be determined by multiplying the normalized coordinates, H_x and H_y, by the maximum x and y field magnitudes:

$$f_x = H_x F_x , \text{ and } f_y = H_y F_y ,$$

where F_x and F_y are the maximum x and y field magnitudes, and f_x and f_y are the field coordinates in field units.
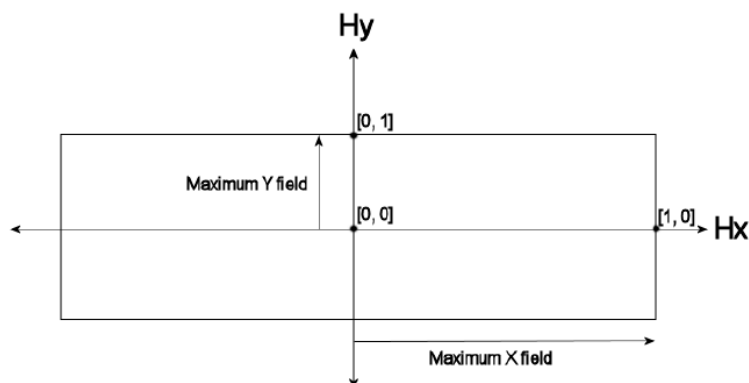
For example, suppose 3 field points are defined in the (x, y) directions using object height in lens units at (0.0, 0.0), (10.0, 0.0), and (0.0, 3.0). The field point with the maximum x coordinate is the second field point, and the maximum x field is therefore 10.0. The field point with the maximum y coordinate is the third field point, and the maximum y field is therefore 3.0. The normalized coordinate (Hx = 0, Hy = 1) would refer to the field coordinates (0.0, 3.0). The normalized coordinate (Hx = 1, Hy = 0) would refer to the field point (10.0, 0.0). The normalized coordinate (Hx = 1, Hy = 1) would refer to the field coordinates (10.0, 3.0). Note the normalized field coordinates can define field coordinates that do not correspond to any defined field point. The maximum x and y field are always positive values.

If a fourth field point at (-10.0, -3.0) were added in the above example, the maximum x and y field values would not change. Normalized rectangular coordinates should always be between -1 and 1.

Normalized pupil coordinates

Normalized pupil coordinates are often used in both the Zemax program and documentation. There are two normalized pupil coordinates: Px and Py. Normalized pupil coordinates are convenient because useful pupil locations may be defined in a manner that does not change with the aperture size or position. For example, the normalized pupil coordinate (0.0, 1.0) is always at the top of the pupil, and therefore defines a marginal ray. The normalized pupil coordinate (0.0, 0.0) always goes through the center of the pupil, and therefore defines a chief ray.

The normalized pupil coordinates represent points on a unit circle. The radial size of the pupil is defined by the radius of the paraxial entrance pupil, unless ray aiming is turned on, in which case the radial size of the pupil is given by the radial size of the stop. For more information on ray aiming see Ray Aiming.

For example, if the entrance pupil radius (not diameter) is 8 mm, then (Px = 0.0, Py = 1.0) refers to a ray which is aimed to the top of the entrance pupil. On the entrance pupil surface, the ray will have a coordinate of (x = 0.0 mm, y = 8.0 mm).

Note that the normalized pupil coordinates should always be between -1 and 1, and that

$$P_x^2 + P_y^2 \leq 1 .$$

A significant advantage of using normalized pupil coordinates is that rays defined in normalized coordinates remain meaningful as the pupil size and position changes. Suppose prior to optimizing a lens design, a ray set is defined to compute the system merit function. By using normalized coordinates, the same ray set will work unaltered if the entrance pupil size or position or object size or position is changed later, or perhaps even during the optimization procedure.

Object space numerical aperture

Object space numerical aperture is a measure of the rate of divergence of rays emanating from the object surface. The numerical aperture is defined as the index of refraction times the sine of the paraxial marginal ray angle, measured in object space. The marginal ray defines the boundary of the cone of light diverging from an object point.

Parameter data

Parameter data values are used to define certain non-standard surface types. For example, parameter data may include aspheric coefficients, grating spacings, or tilt and decenter data. For a complete discussion of the parameter data values see Parameter data.

Paraxial and parabasal rays

The term paraxial means "near the axis". Paraxial optics are optics that are well described by the linear form of Snell's law. Snell's law is:

$$n \sin \theta = n' \sin \theta' .$$

For small angles this becomes

$$n \theta = n' \theta' .$$

Many definitions in optics are based upon this assumption of linearity. Aberrations are deviations from this linearity, and so the paraxial properties of optical systems are often considered the properties the system has in the absence of aberrations. Paraxial rays are traced using formulas which assume the optical surface power is based only upon the vertex radius of curvature, ignoring local linear tilts and higher order curvature of the surface.

Paraxial data is computed on a plane tangent to the surface vertex, assuming the vertex radius of curvature is an acceptable approximation to the surface power over the entire aperture of the surface. Certain unusual surface types do not have a paraxial analog, so the real ray tracing calculations are made at these surfaces, even for a paraxial ray.

Zemax computes many paraxial entities, such as focal length, F/#, focal position, entrance pupil diameter, and others. These values should be used with caution when the optical system has components which violate the assumption that the vertex curvature is an acceptable approximation to the surface power over the entire aperture of the surface.

For many analysis features, paraxial data is required, typically as a reference against which real rays are measured. To ensure these features work properly, even for optical systems that do not meet the paraxial assumption above, Zemax traces "parabasal" rays which are real (real means using Snell's law explicitly) that make small angles with respect to a reference ray, which is usually an axis or chief ray. The parabasal rays are used to compute the limiting properties of the system as the stop size is decreased, which provides a good estimate of the paraxial properties.

The reason Zemax uses parabasal rays rather than paraxial formulas is because many optical systems include non-paraxial components. Non-paraxial means these components are not well described by conventional axial first-order theory. This includes tilted or decentered systems, systems using holograms, diffractive optics, general aspheres, and gradient index lenses.

In summary, paraxial ray data is computed using first order approximations to the surface power for tracing rays, while parabasal rays are real, exact ray traces close to a chief or reference ray. Most paraxial data, such as EFL, F/#, and magnification, use paraxial rays and the data is invalid if the optical system is not well described by the vertex power of every surface. Most analysis features in Zemax use parabasal rays, to allow these features to work with a greater range of optical systems, including those with optical surfaces not well described solely by their vertex surface power.

Paraxial image height

The paraxial radial size of the image in lens units of the full field image at the paraxial image surface.

Paraxial magnification

The radial magnification, being the ratio of paraxial image height to object height. The paraxial magnification is measured at the paraxial image surface. The value is always zero for infinite conjugate systems.

Paraxial working F/#

The paraxial working F/# is defined as

$$W = \frac{1}{2n\tan\theta},$$

where θ is the paraxial marginal ray angle in image space and n is the index of refraction of image space. The paraxial marginal ray is traced at the specified conjugates. For non-axial systems, this parameter is referenced to the axis ray and is averaged over the pupil. The paraxial working F/# is the effective F/# ignoring aberrations. See also the definition for "working F/#".

Primary wavelength

The primary wavelength in micrometers is displayed. This value is used for calculating most other paraxial or system values, such as pupil positions.

Radii

The radius of curvature of each surface is measured in lens units. The convention is that a radius is positive if the center of curvature is to the right (a positive distance along the local z axis) from the surface vertex, and negative if the center of curvature is to the left (a negative distance along the local z axis) from the surface vertex. This is true independent of the number of mirrors in the system.

Real propagation

A real propagation means the rays are propagated in the direction energy would actually flow. See also the definitions for "Virtual propagation" and "Thicknesses".

Sagittal and Tangential

The term "tangential" refers to data computed in the tangential plane, which is the plane defined by a line and one point: the line is the axis of symmetry, and the point is the field point in object space. The sagittal plane is the plane orthogonal to the tangential plane, which also intersects the axis of symmetry at the entrance pupil position.

For typical rotationally symmetric systems with field points lying along the Y axis, the tangential plane is the YZ plane and the sagittal plane is the plane orthogonal to the YZ plane which intersects the center of the entrance pupil.

The problem with this definition is that it is not readily extended to non-rotationally symmetric systems. For this reason, Zemax instead defines the tangential plane to be the YZ plane regardless of where the field point is, and tangential data is

always computed along the local y axis in object space. The sagittal plane is the orthogonal to the YZ plane, and intersects the center of the entrance pupil in the usual way, and sagittal data is always computed along the x axis in object space.

The philosophy behind this convention is as follows. If the system is rotationally symmetric, then field points along the Y axis alone define the system imaging properties, and these points should be used. In this case, the two different definitions of the reference planes are redundant and identical. If the system is not rotationally symmetric, then there is no axis of symmetry, and the choice of reference plane is arbitrary.

One feature, the computation of Fast Semi-Diameters (see the Advanced section of the System Explorer), does use the "true" tangential plane, which Zemax defines as the plane that contains the actual field point and the z axis in object space.

Semi-diameters

The size of each surface is described by the semi-diameter setting. The default setting is the radial distance to the aperture required to pass all real rays without clipping any of the rays. Typing a value for the semi-diameter column results in the character "U" being displayed next to the value. The "U" indicates that the semi-diameter is user defined. When a user-defined semi-diameter is placed on a surface with refractive power (which is done by typing in a value in the appropriate column), and no surface aperture has been defined, Zemax automatically applies a "floating" aperture to the surface. A floating aperture is a circular aperture whose radial maximum coordinate is always equal to the semi-diameter of the surface. For more information on surface aperture types, see the Aperture section of the Surface Properties (in the Lens Data Editor).

Semi-diameters on any surface for axial symmetric systems are computed very accurately, as long as the surface does not lie within the caustic of the ray bundle (note this usually occurs at or near the image surface). Zemax estimates semi-diameters for axial systems by tracing a few marginal pupil rays. For non-axial systems, Zemax estimates the required semi-diameters using either a fixed number of rays or by an iterative technique, which is slower but more accurate. See the explanation of "Fast Semi-Diameters" in Advanced for details. It is important to note that the "automatic" semi-diameter computed by Zemax is an estimate, although it is generally a very good one.

Some surfaces may become so large in aperture that the surface z coordinate becomes multiple valued; for example, a very deep ellipse may have more than one z coordinate for the same x and y coordinates on the surface. For the case of spherical surfaces, this condition is called "hyperhemispheric" and Zemax uses this term even if the surface is not a sphere. Hyperhemispheric surfaces are denoted by an asterisk "*" in the semi-diameter column. The indicated semi-diameter is of the outer edge of the surface, which will have a smaller radial aperture than the maximum radial aperture.

Sequential ray tracing

Sequential ray tracing means rays are traced from surface to surface in a predefined sequence. Zemax numbers surfaces sequentially, starting with zero for the object surface. The first surface after the object surface is 1, then 2, then 3, and so on, until the image surface is reached. Tracing rays sequentially means a ray will start at surface 0, then be traced to surface 1, then to surface 2, etc. No ray will trace from surface 5 to 3; even if the physical locations of these surfaces would make this the correct path.

See also the "Non-sequential ray tracing" definition.

Special characters

There are many places in Zemax where user defined file, material, glass, or other names may be provided. Generally, Zemax allows any characters to be used in these names, except for a few reserved "special characters" The special characters are space, semi-colon, single quote, and tab.

Strehl ratio

The Strehl ratio is one commonly used measure of optical image quality for very high quality imaging systems. The Strehl ratio is defined as the peak intensity of the diffraction point spread function (PSF) divided by the peak intensity of

the diffraction point spread function (PSF) in the absence of aberrations. Zemax computes the Strehl ratio by computing the PSF with and without considering aberrations, and taking the ratio of the peak intensity. The Strehl ratio is not useful when the aberrations are large enough to make the peak of the PSF ambiguous, or for Strehl ratios smaller than about 0.1.

Surface apertures

Surface apertures include circular, rectangular, elliptical, and spider shaped apertures which can vignette rays. There are also user defined shapes for surface apertures and obscurations; and a "floating" aperture that is based upon the current semi-diameter value. Surface apertures do not affect ray launching or tracing, except for the termination of a ray if it does not pass the surface aperture. Surface apertures have no effect on the system aperture. For more information, see the Aperture section of the Surface Properties

System aperture

The system aperture is the overall system F/#, Entrance Pupil Diameter, Numerical Aperture, or Stop Size. Any of these 4 quantities is sufficient to define the other 3 for a particular optical system. The system aperture is used to define the object space entrance pupil diameter, which in turn is used to launch all rays. The system aperture is always circular. Rays may be vignetted after being launched by various surface apertures. There is only one system aperture, although there may be many surface apertures.

Thicknesses

Thicknesses are the relative distance to the next surface vertex in lens units. Thicknesses are not cumulative, each one is only the offset from the previous vertex along the local z axis. The orientation of the local z axis can change using coordinate breaks (see Coordinate Break) or surface tilts and decenters (see the Tilt/Decenter section of the Surface Properties).

Thicknesses corresponding to real propagation (see "Real propagation") always change sign after a mirror. After an even number of mirrors (including zero mirrors), thickness are positive for real propagations and negative for virtual propagations (see "Virtual propagation"). After an odd number of mirrors, thicknesses are negative for real propagations and positive for virtual propagations. This sign convention is independent of the number of mirrors, or the presence of coordinate breaks. This fundamental convention cannot be circumvented through the use of coordinate rotations of 180 degrees.

Total internal reflection (TIR)

TIR refers to the condition where a ray makes too large an angle with respect to the normal of a surface to meet the refraction condition as specified by Snell's Law. This usually occurs when a ray with a large angle of incidence is refracting from a high index media to a lower index media, such as from glass to air. When doing sequential ray tracing, rays which TIR are considered errors, and are terminated. Physically, the ray would reflect rather than refract from the boundary, but Zemax does not consider this effect when doing sequential ray tracing. For non-sequential ray tracing, rays which TIR are properly reflected.

Total track

Total track is the length of the optical system as measured by the vertex separations between the "left most" and "right most" surfaces. The computation begins at surface 1. The thickness of each surface between surface 1 and the image surface is considered, ignoring any coordinate rotations. The surface which lies at the greatest z coordinate defines the "right most" surface, while the surface with the minimum z coordinate defines the "left most" surface. Total track has little or no value in non-axial systems.

Vignetting factors

Vignetting factors are coefficients which describe the apparent entrance pupil size and location for different field positions. Zemax uses five vignetting factors: VDX, VDY, VCX, VCY, and VAN. These factors represent decenter x, decenter y, compression x, compression y, and angle, respectively. The default values of all five factors are zero, which indicates no vignetting.

Both the field of view and the entrance pupil of an optical system can be thought of as unit circles. The normalized field and pupil coordinates, defined in "Normalized field coordinates", are the coordinates on these two unit circles. For example, the pupil coordinates (px = 0, py = 1) refer to the ray which is traced from some point in the field to the top of the entrance pupil. If there is no vignetting in the system, Zemax will trace rays to fill the entire entrance pupil during most computations.

Many optical systems employ deliberate vignetting. This means a portion of the rays are intentionally "clipped" by apertures other than the stop surface. There are two common reasons for introducing vignetting in an optical system. First, vignetting decreases the size of the lenses, particularly in wide angle lenses. Second, vignetting may remove a portion of the beam which would be excessively aberrated. Vignetting usually increases the F/# as a function field angle (which darkens the image), but the image quality may improve if the most severely aberrated rays are clipped.

Vignetting factors redefine the entrance pupil for a specific field position. The normalized pupil coordinates are modified using two successive transformations. First, the coordinates are scaled and shifted:

$$P'_x = VDX + P_x(1 - VCX), \text{ and}$$

$$P'_y = VDY + P_y(1 - VCY), .$$

The scaled and shifted coordinates are then rotated by the vignetting angle:

$$P''_x = P'_x \cos\theta - P'_y \sin\theta, \text{ and}$$

$$P''_y = P'_x \sin\theta + P'_y \cos\theta,$$

where θ is the vignetting angle VAN. The VDX term can shift the apparent pupil left and right, while VCX makes the pupil larger or smaller in the x direction. Similar results hold for the VDY and VCY values. Note that if the vignetting factors are all zero, the pupil coordinates are left unmodified. Vignetting factors provide a convenient way of designing optics which employ vignetting. However, there are restrictions to using vignetting factors that must be understood.

Some Zemax features are capable of tracing rays from arbitrary field positions where no vignetting factors have been assigned. These features may not provide completely accurate results for data computed at field positions other than the defined fields. Some features will remove the vignetting factors for these computations by placing a clear aperture on each surface that vignettes the rays an equivalent amount. Features that automatically remove the vignetting factors are described in the Analysis chapter.

Some features in Zemax do not automatically remove vignetting factors for intermediate field positions, such as ray operands in the merit function (operands like REAX that can launch a single ray, for example) or ZPL macros. If the vignetting factors are not removed, Zemax will attempt to interpolate the vignetting factors. For rotationally symmetric systems, or systems with field points entirely along the y axis, Zemax interpolates between adjacent field points to estimate the vignetting factors to use at intermediate field points. For more general optical systems with X field values, Zemax uses the closest defined field point for determining the vignetting factors for an arbitrary field point.

Once a vignetting factor is defined, it is up to the designer to ensure that rays beyond the apparent pupil are in fact vignetted! If the vignetting factor is used to shrink the size of the lenses, then the lenses should be made no larger than is required to pass the rays which are at the edge of the apparent pupil. If rays from beyond the vignetted aperture are allowed to pass in the real optical system, then the lens performance will not correlate with the computer model.

Identical or nearly identical field coordinates may not be defined with different vignetting factors. Field coordinates must be different by roughly 1E-06 times the maximum field coordinate if two neighboring fields use different vignetting factors. This is required because Zemax must determine vignetting factors for any field coordinates, not just those at defined field positions; and identical field coordinates with different vignetting factors have no physical interpretation. The proper way to set up this sort of system is to use multiple configurations, and change the vignetting factors via the multi-configuration editor.

The vignetting factors work with and without ray aiming turned on. If ray aiming is off, then the paraxial entrance pupil is remapped according to the equations given earlier. If ray aiming is turned on, then the remapping is done at the stop surface.

One possible application of vignetting factors is to account for pupil aberration without using the ray aiming feature. This is an advanced trick which can be used to speed up ray tracing in wide angle systems.

Vignetting factors may be defined on the "Field Data" dialog box. See the Fields section of the System Explorer for more information. Vignetting factors may also be zoomable parameters; see the Chapter "Multi-Configurations". For more information on the use of vignetting as a design tool, see any of the good books referenced in the first chapter.

Virtual propagation

A virtual propagation means the rays are propagated in a direction opposite to the direction energy would actually flow. Virtual propagations are often useful for placing virtual sources or pupils. See also the definitions for "Real propagation" and "Thicknesses".

Wavelength data

Wavelength data are always measured in micrometers referenced to "air" at the current system temperature and pressure. The default system temperature is 20 degrees Celsius, and the default air pressure is 1.0 atmospheres. If the system temperature and/or pressure is modified, or under the control of multi-configuration operands, care must be taken to adjust the wavelengths to the new air temperature and pressure.

Wavelength data is entered in the Wavelengths section of the System Explorer.

Wavelength data are always measured in micrometers referenced to "air" at the system temperature and pressure.

Working F/#

Working F/# is defined as

$$W = \frac{1}{2n\sin\theta},$$

where θ is the real marginal ray angle in image space and n is the index of refraction of image space. The marginal ray is traced at the specified conjugates. Working F/# ignores surface apertures but considers vignetting factors.

For off axis field points or non-axial systems, working F/# is determined by the average of the square of the numerical aperture between the axis ray and four marginal rays, at the top, bottom, left, and right side of the vignetted pupil. The average of the square of the numerical aperture of the four rays is converted back to equivalent F/#.

Working F/# is generally much more useful than image space F/# because it is based upon real ray data at the actual conjugates of the lens. See also the definition for "Paraxial working F/#".

If the marginal rays cannot be traced (due to ray errors) then a smaller pupil is temporarily used to estimate the working F/#. In this case, Zemax scales the data to estimate the working F/# at the full pupil size, even though rays are not be traceable at the full aperture.

If the marginal rays are nearly parallel to the chief ray, the resulting F/# may become so large as to be inaccurate. Zemax will automatically "cap" the F/# at 10,000 if the calculated F/# becomes larger than 10,000. This result simply means the F/# cannot be calculated accurately using rays. Such a large F/# means the output beam is nearly collimated, and many assumptions Zemax makes are not valid in this case. The two solutions are to bring the nearly collimated beam to a focus using a paraxial lens, or to use the exit pupil size and position to estimate the F/#. For more information, see the explanation of "Method To Compute F/#" in the Advanced section of the System Explorer.

For a generalized definition of F/# useful for evaluating image brightness, see the discussion of "Effective F/#" in the discussion about the Relative Illumination analysis.

In-Line Help

This is the In-Line Help section of the Help Files. Click the "+" next to the "In-Line Help" header in the Contents pane to expand and navigate through this section.

File Tab

This tab provides access to all of OpticStudio's File management capabilities

---
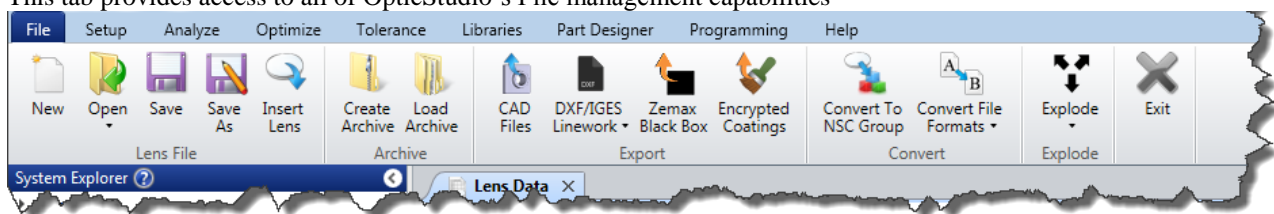
Lens File Group

These provide the standard Windows file management functionality. The Open icon gives acces to the File Open dialog as well as the most recently used files list.

All OpticStudio data is stored in {filename}.zmx, and OpticStudio also creates {filename}.ses and {filename}.cfg to allow any open Analysis windows to open in the same state as they were in when the file was saved.

The Insert Lens Icon Inserts surface data from a previously stored lens data to the current lens file.

New/Open/Save/Save As

These provide the standard Windows file management functionality. The Open icon gives acces to the File Open dialog as well as the most recently used files list.

All OpticStudio data is stored in {filename}.zmx, and OpticStudio also creates {filename}.ses and {filename}.cfg to allow any open Analysis windows to open in the same state as they were in when the file was saved.

Insert Lens

Inserts surface data from a previously stored lens data to the current lens file. This option is similar to the "Open" command, but the current lens data is not disturbed. After the file to insert is selected, OpticStudio will prompt for the surface number where the new lens data should be inserted. New surfaces will be inserted into the lens prescription to make room for the new data.
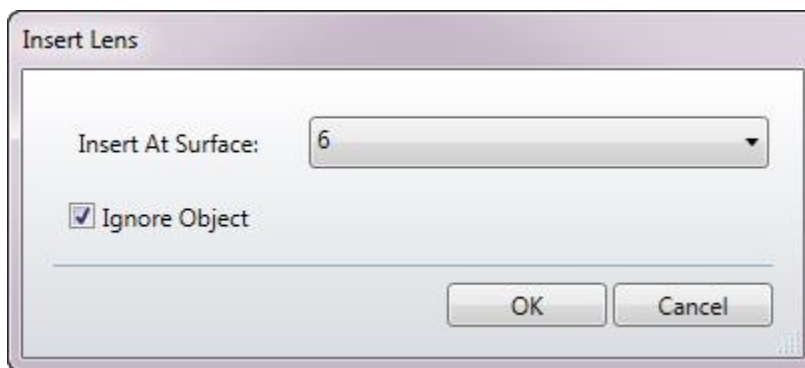
The dialog box also has an "Ignore Object" checkbox which when selected will ignore the object thickness of the lens being inserted.

Archive Group
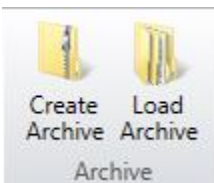
The main "lens" file in Zemax ends in the extension ZMX. However, many other files are used to define critical data for the optical design. These files include glass and gradient index data catalogs, coating data files, user defined aperture files, CAD format files, bulk and surface scatter data, externally defined DLL files, ZPL macros, and many others. The possibly large number of these data files, and the fact that these files may be stored in a wide range of folder locations, makes backing up and archiving all the individual files cumbersome and error prone. The same problem occurs if the lens design needs to be sent to a colleague via email. The files must all be collected, backed up, and then stored or sent. To restore the complete design and all the referenced files requires all the files be restored to the correct folder so Zemax can find the files.

This feature solves this data backup and restoration problem. The backup process gathers all of the data and files used by the design, including all data and files used by any analysis windows open at the time the backup archive is created, and stores the data in a single archival file. The Zemax Archive format (ZAR) is a one-file solution to making a complete backup of the optical design.

The archive works by storing a complete copy of each file the design uses in the ZAR file. The original folder information is not stored, just the file type, name, and data. For example, the ZAR will contain any AGF glass catalogs

the design was using. When the design is restored from the ZAR file, the AGF file will be created and placed in the current folder for glass catalogs. This procedure allows the ZAR to be restored to another user's machine that may have different drive and folder names for the glass catalogs than the machine used to create the ZAR was using.

Create Archive

When the "Create Archive" feature is selected, Zemax first prompts for the file name to create. This may be an existing file or a new name may be entered. The ZAR file defaults to the current lens file name, but may be given any valid name. The file name does not need to match the lens file name, however, if the ZAR file is renamed, it is important to understand that when restored the original lens file name is restored. When the file name is selected, a dialog box appears with the options "OK" and "Cancel".
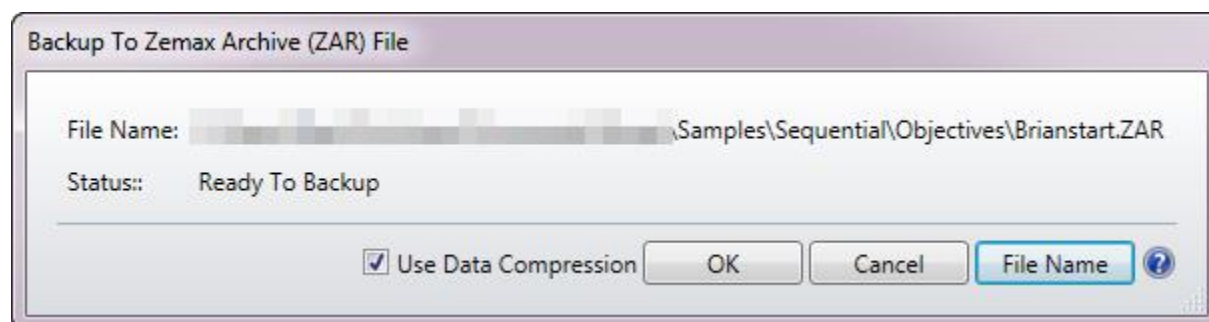
Select OK to start the archival process or Cancel to abort the archive.

Note that the amount of time it takes to create the ZAR depends upon how many files need to be backed up, how large those files are, and whether data compression is applied or not. Most files used by Zemax are modest in size, but some files, such as ZRD, source ray DAT, and CAD files are potentially quite large. If any individual file exceeds 50.0 MB in size, Zemax will prompt the user to decide if the file should still be placed in the archive file. Note if "No" is selected the file WILL NOT be archived and will not be able to be restored from the ZAR file. In this case a separate backup must be made as the file will not be stored in the archive. Note also that the individual part files associated with a CAD Assembly object (e.g. see "CAD Assembly: Autodesk Inventor" on page 460) are not stored in the archive. Thus, a separate backup of those individual part files must be made and sent along with the ZAR when providing a colleague with the archive of a design containing a CAD Assembly object.

The Create Archive feature also supports optional data compression. This data compression is applied to the archive to yield a smaller file size. Older versions of Zemax may not have the ability to read these compressed files, and so the use of data compression may be disabled to provide compatibility with older versions of Zemax if required. The data compression algorithm used by Zemax is designed to be compatible with future versions of Zemax and provides reasonably good compression with little overhead. However, the algorithm used does not generally provide the highest possible levels of compression. Compression is not applied to ZRD files, nor any mother files larger than 50.0 MB in size, as the compression time becomes unreasonably long on these large files.

At the end of the archive process, the status box will report how many files were saved. Press "Close" to close the archive feature. Once the ZAR is generated, the file may be renamed, backed up, or sent to a colleague, like any other data file.

Load Archive

This feature opens a previously created archive file in ZAR format and (optionally) restores all the files contained in the archive.
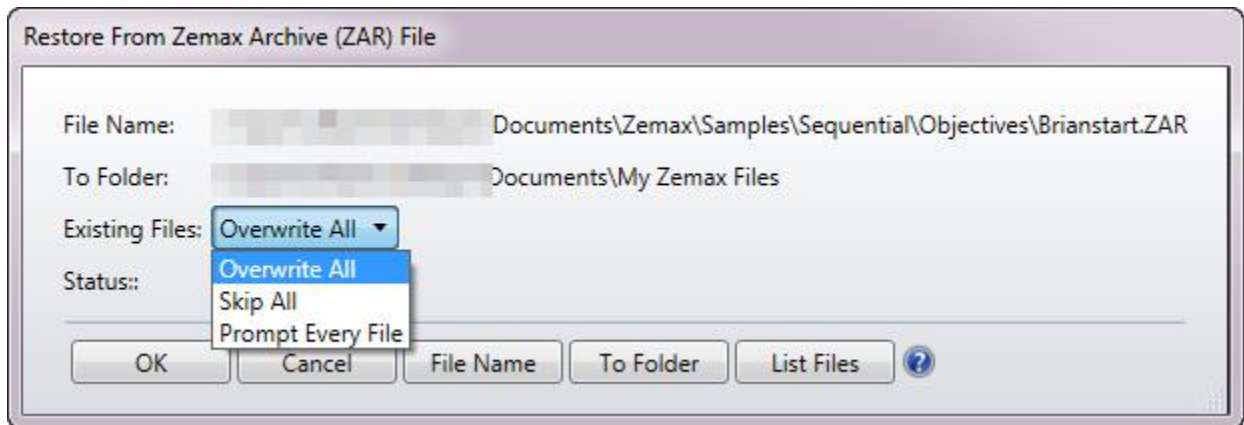
When this feature is selected, Zemax first prompts for the ZAR file name to restore. The dialog above then appears

To Folder: This button allows the user to select the master folder in which the archived files will be restored. The initial default folder is the "Lens Folder" However, after the first archive file is restored, the most recently used folder will be the default, but any valid folder for which the user has read and write permission can be selected. This is the folder where the main Zemax file (*.ZMX), config file (*.CFG), session file (*.SES), and any ray database (*. ZRD) files are stored. Most other files, such as glass catalogs (*.AGF) and Dynamic Link Libraries (*.DLL) are stored in the folders the current running Zemax is using, as defined by the Project Preferences.

Existing Files: This feature tells OpticStudio what to do if the archive contains a file of the same name as an existing file. The options are Overwrite All (the file in the ZAR will overwrite the existing file without warning or saving the old file), Skip All (the existing files will be retained and the files in the ZAR will be ignored), and Prompt Every File (OpticStudio will ask you whether to overwrite or skip every single file when a duplicate exists).

OK: Begins the restoration of files from the ZAR.

Cancel: Exits the dialog without restoring the files from the ZAR.

List Files: Creates a window that lists all of the files in the ZAR archive, along with the file size and date. If the file is compressed, the compressed file size and not the original file size is listed.

When the restoration is complete, press Close to close the dialog box. OpticStudio will then immediately load the ZMX file that was just restored.

Export Group
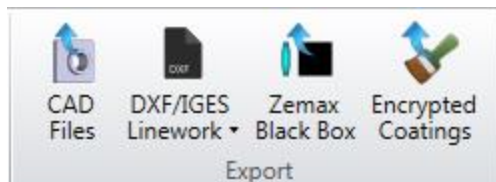
The group of icons provides access to exporting data in STEP, IGES, SAT and STL solid CAD format, and as linework DXF/IGES files for older CAD software. It also allows a sequential Zemax file to be exported as as 'Black Box' file which can be given to a customer as a file that ray-traces perfectly but does not allow the design data to be seen by the user. Enctypted Coatings provode a similar capability

CAD Files

Exports the current lens data to a file as a collection of 3D solids in IGES, SAT, STEP, or STL format.
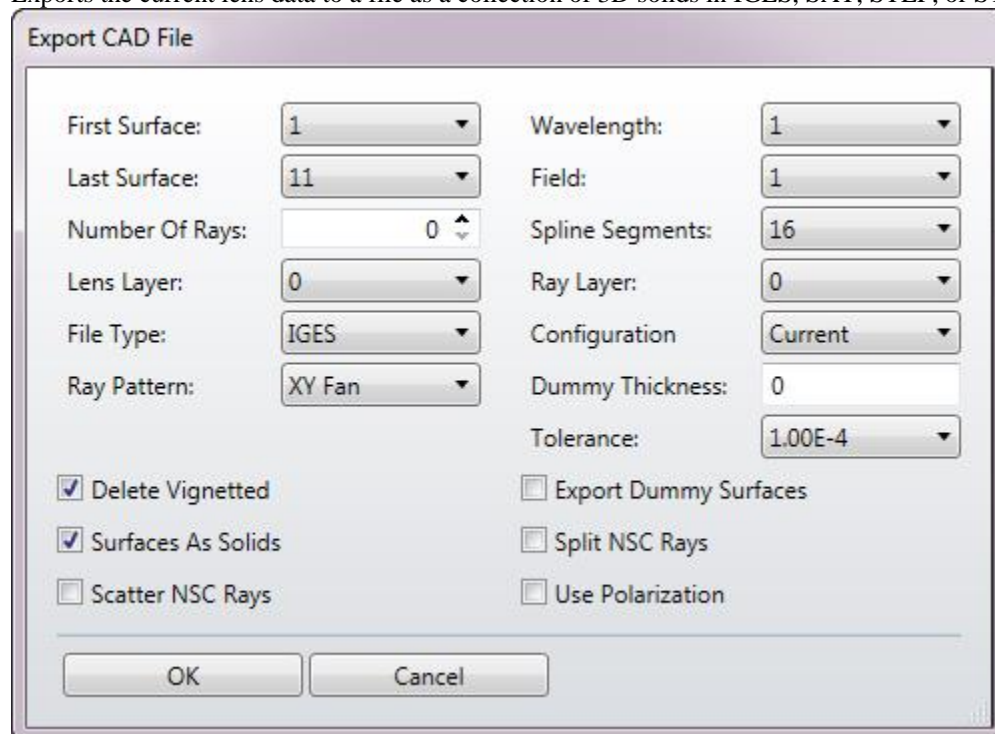
First/Last Surface or Object: The range of surfaces to include in the exported data. In Non-Sequential mode, these options are First and Last Object.

Wavelength: The wavelength number to use for any traced rays to be exported.

Field: The field number to use for any traced rays to be exported.

Number of Rays: The number of rays to trace; the exact meaning depends upon the "Ray Pattern". When exporting to STL format, no rays are exported as STL does not support a line entity.

Split NSC Ray s , Scatter NSC Rays, Use Polarization: These settings only affect rays from NSC sources. For more information, see the discussion on the similar controls described in "NSC 3D Layout" on page 131.

Ray Pattern: Selects the type of ray set to export. This control is very similar to that defined for the 3D layout plot. The solid beam option exports a solid volume representing the envelope of the rays. The solid beam option is only available in sequential mode, and required the the marginal rays to be all traceable and unvignetted. The marginal rays may not form a caustic at any surface. Solid beams are not generated for non-sequential surfaces. The number of rays must be at least 8 to export a solid beam, and a smoother solid results if more rays are used.

Lens/Ray Layer: Selects the layers of the output file to place lens and ray data.

Configuration: One or more configurations may be exported at once using this control. Current will export only data from the current configuration. Selecting a configuration number, such as 1, 2, or 3, will export data only from that configuration.

"All By File" will export all configurations, but each to a separate file name. The file name for each configuration is the specified save file name (the save file name is selected after OK is pressed) with the name "_config000x" appended to indicate the data is for configuration "x".

"All By Layer" will export all configurations, but each to a separate layer within the same file. The first configuration will be placed in the layer specified by the lens layer and ray layer settings. Subsequent configurations are placed in the lens and ray layers incremented by 1 for each configuration. For example, if the lens layer is 1 and the ray layer is 10 and there are 3 configurations, then configuration 1 will have lens data in layer 1 and ray data in layer 10. Configuration 2 will have lens and ray data in layers 2 and 11, and configuration 3 will have lens and ray data in layers 3 and 12, respectively.

"All At Once" will export all configurations at once in a single layer. No attempt is made by Zemax to avoid overlapping or redundant data export.

File Type: Selects the output file format type as one of STEP, IGES, SAT and STL formats

Delete Vignetted: If checked, rays are not included in the data export if they will be vignetted by any surface.

Surfaces As Solids: This control only applies to sequential surfaces, and not NSC surface objects. If checked, surfaces on either side of a glass material will be combined to form a single solid object, where possible. Not all Zemax surface types may be combined to form a single solid. Both surfaces must have the same aperture shape to export surfaces as solids. For example, both surfaces must have circular apertures, or both surfaces must have rectangular apertures. If the aperture types do not match then the surfaces are exported as single surfaces.

Note that the first surface may not be the same as the last surface if this option is enabled. If it is, Zemax will default to exporting the entire lens system. To export one lens, the first surface should be the front surface and the last surface should be the back surface.

Export Dummy Surfaces: If checked, dummy surfaces will be exported. A dummy surface is a surface that does not reflect or refract rays.

Dummy Thickness: The distance in lens units for the thickness of dummy surfaces if they are automatically converted to solids. Both "Export Dummy Surfaces" and "Surfaces As Solids" need to be checked for this control to have any effect.

Tolerance: The *approximate* accuracy of the exported objects in lens units. Smaller tolerances yield more accurate representations at the expense of larger exported files and longer computation times.

Spline Segments: The number of segments to use when spline entities are exported.

General Points to Note

The Initial Graphics Exchange Specification (IGES) is an American National Standard whose intended purpose is to facilitate transfer of data between CAD programs. OpticStudio currently supports version 5.3 of the IGES standard. For more information on IGES, contact U.S. Product Data Association, P. O. Box 3310, Gaithersburg, MD 20885-3310. The SAT format is used by the ACIS geometry modeling engine developed by Spatial Technologies. The STEP format is AP203 as defined by ANS US PRO/IPO-200-203-1994, also available from the U.S. Product Data Association.

OpticStudio exports five types of data:

> **Lines:** Lines are used to indicate the paths of rays traveling through the optical system. Within GRIN media, rays are exported as a series of line entities. The STL format does not support lines, so rays are not exported.

> **Surfaces:** Surfaces may be of arbitrary shape, including user defined surfaces; with any aperture shape Zemax supports, including user defined apertures. Data for some types of surfaces (particularly aspheres and toroids) which are exported as splines have an accuracy that depends on the number of spline points used; more points

yields more accuracy at the expense of larger, slower exported data files. For more information on the limitations of the accuracy of the exported objects see the discussion below.

Lens Solid: Lens Solids include a non-zero volume enclosed by two surfaces, with the edge region formed by the extrusion of the edges of the two surfaces. Most surfaces surrounding glass with similar shaped apertures (e.g. both surfaces have rectangular apertures) may be exported as solids.

Faceted Solid: A Faceted Solid is defined by a collection of triangles that form a volume, such as a prism or Fresnel lens. Non-sequential STL and POB objects, among others, are exported as Faceted Solids.

Parametric Solid: Some NSC objects, such as the Torus Volume, are exported as exact solid NURBS objects.

OpticStudio can export the surface or solid data for all sequential surfaces and non-sequential objects. Zemax exports all surfaces and rays in a 3D coordinate system referenced to the global coordinate reference surface.

If exporting surfaces as solids, OpticStudio will export sequential dummy surfaces as thin solids. The front and back face will be of the shape of the surface, with a thickness equal to the dummy thickness in lens units. Mirror surfaces are exported according to the settings for the mirror substrate.

Most NSC objects are exported exactly, or by using one or more spline based surfaces similar to those used for sequential surface export. However, some objects, such as user defined objects, are exported using a faceted approximation to the smooth surfaces. Increasing the drawing resolution for these objects will also increase the resolution of the exported representation.

STL is a faceted format, and all STL surfaces or objects objects are exported as faceted approximations.

This feature will prompt for the file name and path to save the exported data to. The default file name and folder are the same as the current lens file, with an extension that depends upon the CAD file type format selected.

Limitations of exported data

OpticStudio supports many types of complex surface and object shapes. Within OpticStudio, these shapes are modeled exactly, yielding the very high numerical precision required for optical ray trace analysis. However, most of the complex optical shapes OpticStudio supports have no analog in commonly used CAD programs, and no exact representation in common CAD file formats such as IGES, STEP, and SAT.

For this reason, OpticStudio must approximate the exact surface shape with the closest available analog within the CAD format. This approximation is usually very good, but for some extremely precise aspheric surface shapes, the approximation is not sufficiently accurate for ray tracing. This is not a limitation of OpticStudio. This is a limitation of the CAD data exchange format and the CAD programs OpticStudio is being asked to export data to. In these rare cases, it may be necessary to recreate the complex geometry within the CAD program directly rather than use the OpticStudio export capability.

There is always the possibility that the conversion from the OpticStudio representation of surfaces and objects to the CAD file format representation will cause a loss of precision.

DXF/IGES Linework
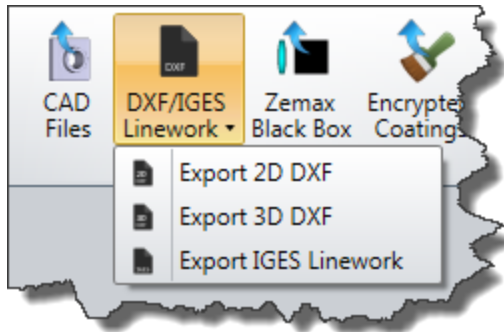
This set of features is intended for older CAD programs that cannot handle 3D solid export. The lineworks produced are not intended for manufacturing purposes. Only a limited set of sequential surfaces are supported for Linework output. The Export CAD feature should be used unless there is a compelling need for these older formats.

Export 2D DXF

Exports the current lens data as a 2D DXF format file. Only available for spherical symmetric lenses.
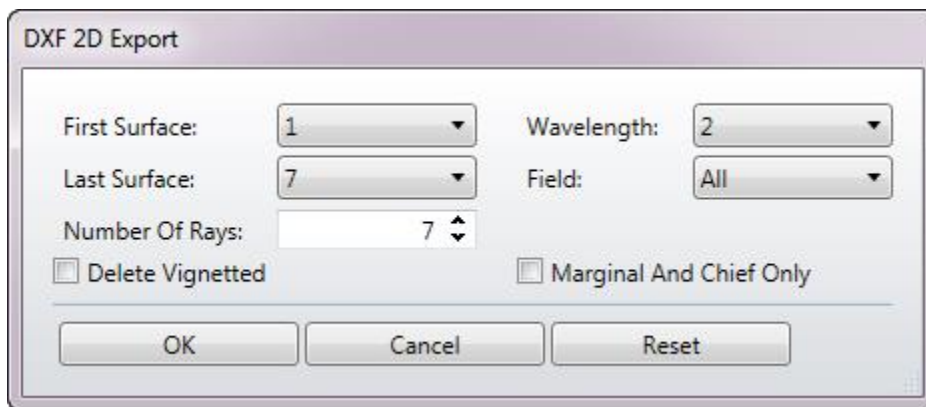
First/Last Surface: The range of surfaces to include in the exported data.

Wavelength: The wavelength number to use for any traced rays to be exported.

Field: The field number to use for any traced rays to be exported.

Number of Rays: The number of rays to trace and include in the export.

Delete Vignetted: If checked, rays are not included in the data export if they will be vignetted by any surface.

Marginal and Chief Only: Draws only the marginal and chief rays, overriding the other ray settings.

This feature will prompt for the file name and path to save the exported data to. The default file is EXPORT.DXF in the <objects>\CAD Files folder. The output will be in DXF format no matter what extension is chosen.

Export 3D DXF

Exports the current lens data as a 3D DXF format file. This feature is similar to the 2D DXF export feature described in the previous section. The key difference is that the 3D DXF supports systems that are not rotationally symmetric.

First/Last Surface: The range of surfaces to include in the exported data.

Wavelength: The wavelength number to use for any traced rays to be exported.

Field: The field number to use for any traced rays to be exported.

Radial Segments: The number of radial segments used to approximate the lens shapes. Larger numbers require more processing time.
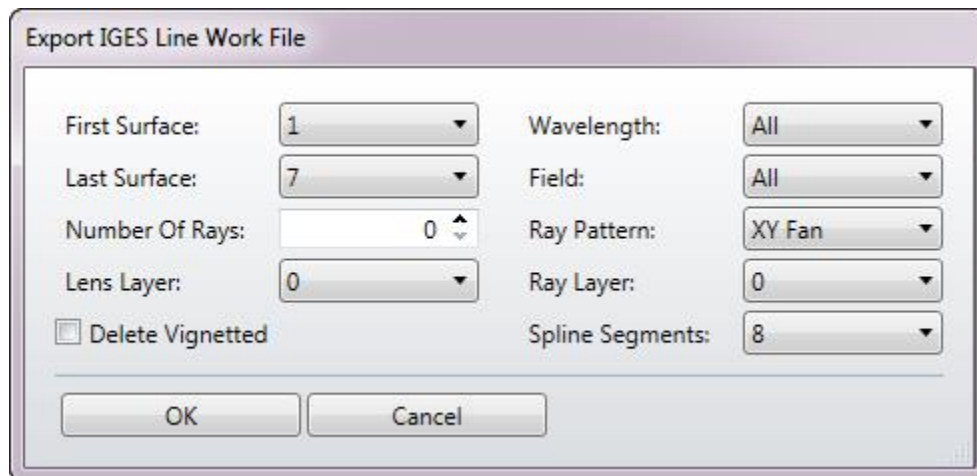
Angular Segments: The number of angular segments used to approximate the lens shapes. Larger numbers require more processing time.

Number of Rays: The number of rays to trace.

Ray Pattern: Choose XY Fan, X Fan, Y Fan, Ring, List, Random, or Grid to indicate what the pattern of rays to be traced should be. The List option indicates that the rays to be traced are user defined and listed in a file, see "Raylist file format" on page 126 for information on the ray list format. If List is selected the Number of Rays setting is ignored. Delete Vignetted If checked, rays are not included in the data export if they will be vignetted by any surface.

Export IGES Linework

Exports the current lens data as an IGES line work format file

First/Last Surface: The range of surfaces to include in the exported data.
Wavelength: The wavelength number to use for any traced rays to be exported.
Field: The field number to use for any traced rays to be exported.
Number of Rays: The number of rays to trace; the exact meaning depends upon the "Ray Pattern".
Ray Pattern: Selects the type of ray set to export. This control is very similar to that defined for the 3D layout plot.
Lens/Ray Layer: Selects the layers of the output file to place lens and ray data.
Delete Vignetted: If checked, rays are not included in the data export if they will be vignetted by any surface.
Spline Segments: The number of segments to use when spline entities are exported.

OpticStudio exports lines, spherical arcs, and splines to represent the shape and position of each surface. OpticStudio does not export the "edge" of the lens, and makes no distinction between dummy surfaces, glass surfaces, and mirrors.

Zemax Black Box

This feature exports data from one or more surfaces in the Lens Data Editor into a "black box" file. The file can then be later applied to a Black Box surface to be reimported into OpticStudio without revealing the details of the design.
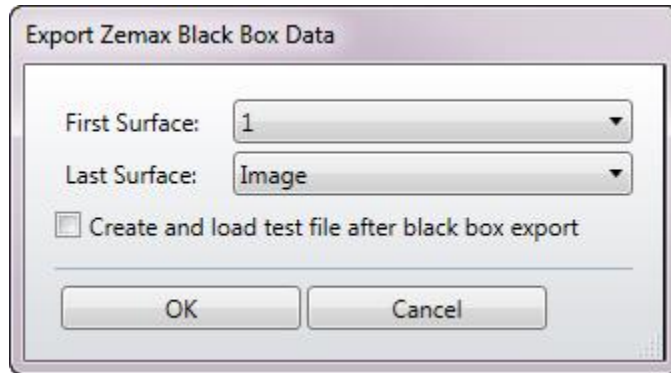
First Surface: The first surface in the range of surfaces to export.
Last Surface: The last surface in the range of surfaces to export.
Create and load...: If this option is checked, the current file will be modified to use the newly created Black Box file. Use of this option is highly recommended to confirm the Black Box reproduces the desired analysis data.

General Points to Note

Optical designers may under certain circumstances be hesitant to divulge the details of their optical designs. This is especially true when significant time and cost is involved in creating the design, and the designer may not be compensated for the work until the end customer has verified the design will perform as specified. This creates a dilemma, because sending the complete ZMX file to a customer allows the customer to verify the design performance, but also reveals in full detail the design itself. This feature is designed to satisfy the needs of both the designer and the end customer in these circumstances.

A Zemax Black Box is an encrypted file that contains the prescription data for a range of surfaces. The file contains most all of the data Zemax normally associates with a surface, including the radius, thickness, aperture, coordinate, dispersion, parameter, and extra data. The file is created by the original designer that holds the intellectual property rights. The file is then sent, usually as part of a complete "playback" Zemax lens file, to the end customer. The encrypted data appears as a "black box" in the Lens Data Editor. Rays go in, and the rays come out, exactly as if the original surfaces and design data were present in the Lens Data Editor. However, the detailed data is actually hidden inside the Black Box, and the user cannot review or modify any of the data contained within the hidden Black Box section.

To create a Black Box file, highlight the desired range of surfaces in the Lens Data Editor, and then select the Export Black Box icon in the Export group of the File Tab. The surface range may also be selected from the Export Black Box Data dialog control directly. The only other option is a check box to "Create and load test file after black box export". *Use of this latter option is highly recommended*, as it allows confirmation that the Black Box was created properly and yields the identical ray tracing and analysis results. The "Create" option will create a test lens file with the same name as the original file, except the letters "_BB" are appended to the end of the name. If a file with the same name as the test file already exists, it will be automatically overwritten. The black box data itself is created in a file with the extension ZBB, and this file is placed in the folder <data>\BlackBoxes. This newly created file will be referenced on the Black Box Lens surface in the Lens Data Editor, and will replace the original surface data in the editors. The name of the Black Box file may not exceed 32 characters, including the .ZBB extension.

Once the Black Box and test playback file are created, it is recommended that the archive feature be used to make a single archival file that can be sent to the end user. See Create Archive

For more information on the Black Box surface, see "Black Box Lens".

The Black Box data structure is designed so that no OpticStudio feature can read or query the prescription data of the surfaces within the Black Box. Rays enter the Black Box, and the rays are traced using the data defined within, but no feature is able to access the curvature, thickness, glass type, aspheric constants, or other data. The refractive lenses in the Black Box are modeled using the full dispersion data from the original file, and thus can be "played back" at any wavelength. If a playback wavelength exceeds the limits of the dispersion data, an error is issued. If any ray is vignetted or TIRs inside the Black Box, it is treated as a "ray miss" error. Note that some features, such as those that compute the weight of a lens, will not be able to access the Black Box data either and will thus return meaningless data for Black Box Lens surfaces.

The black box is drawn on layout plots as a cylinder with a radial aperture defined by the semi-diameter of the first surface in the selected range of surfaces when the black box was created, or by the largest radial aperture of any surfaces within the selected range, whichever is larger. The length of the black box as drawn is equal to the sum of the lengths of the surfaces within the selected range.

There are some restrictions on the type of data that may be exported to a Black Box file:

- Neither the first nor the last surface of the specified range may be ignored.
- If the first surface follows an even number of mirrors, so must the last surface. If the first surface follows an odd number of mirrors, so must the last surface. Neither the first nor the last surface may be mirrors. The range of surfaces may be intended to be used in a mirror space or not, however, the playback will only work if the file is properly set up after either an even or odd number of mirrors as was the set up for the original file.
- Coordinate Break surfaces and surface tilts and decenters are permitted, however the last surface must be oriented in the same coordinate system as the first. This means the local X, Y, and Z axis of the first and last surfaces must all be parallel, and the X and Y decenters must be zero (both limits are tested to a fine tolerance set internally by OpticStudio).
- Not all surface types are supported. Nonsequential, Birefringent, Paraxial, and surfaces that use external data, such as user defined DLL surfaces and grid sag surfaces, and a few other special surface types are not supported. If a need arises for a surface type that is not supported, contact technical support for assistance.
- The stop cannot be placed in between the first and the last surfaces, although the stop may be either of these surfaces. To export lens data on either side of the stop surface, use two (or more) Black Boxes. Any number of Black Box Lens surfaces may be used in the playback system.
- If any coatings are applied to the surfaces within the Black Box, they must be either standard coatings with no tapers, or encrypted coatings (see "Export Encrypted Coating" on page 283). No ideal or table coatings are supported. The export to Black Box feature will automatically convert all non-encrypted coatings to encrypted coatings. The coating name will be changed by prepending "ZEC_" to the existing name. For example, the coating named BB_AR_5 will be exported as an encrypted coating file with the name ZEC_BB_AR_5. If this file already exists, it will be overwritten without warning. The encrypted coating data is not stored within the ZBB file, the data must be included separately when sent to the end user. By far the easiest and most reliable way to do this is to send the resulting Black Box lens system with the coatings in an Archive (ZAR) file.
- Apertures must be placed on all surfaces with power, so that when used the black-box file will correctly clip rays outside of the aperture of the lens. For a tool that converts semi-diameters to apertures see "Convert Semi-Diameters to Floating Apertures" on the Lens Data Editor Toolbar, or apertures can be set by hand using the Aperture tab of the Properties Inspector in the Editor.
- User-defined aperture data files (.UDA) are not supported.
- Note any rays vignetted by the black box surfaces are treated as ray miss errors in the playback system, and this can cause some features to not work properly with certain optical systems, such as telescopes with central obscurations.
- Features that require access to specific surface data, like Physical Optics Propagation, cannot reproduce the same results from a black box file as the original file can.
- Only data for the current configuration is exported. Instead, in a multi-configuration system, each configuration must be exported separately. Upon loading into the playback system, the file name of the ZBB files used can be changed for each configuration using the MCOM operand) in the multi- configuration editor if multiple-configuration control of the black-box system is required.

- Any data controlled by solves are replaced by their current fixed values. No solves are supported inside the Black Box surface. If "Create and load" is selected any solves for surfaces outside the newly created blackbox that reference data now within the blackbox are also turned off.
- Model and Table glasses are not allowed - all dispersion data must be defined in the current glass catalogs. The dispersion data will be extracted from the glass catalog and placed in the black box file. The exception is for gradient index surfaces as described below.
- Index data for gradient index surfaces is generally defined in parameter data that is encrypted within the black box. Gradient 5 surfaces, when converted to a black box, store the gradient polynomial parameters in encrypted form, but the dispersion data remains as defined in the unencrypted SGRIN.DAT file normally installed with OpticStudio. Gradient 6 surfaces, which use dispersion data defined in the GLC.DAT file, also use the unencrypted data in that file normally installed with OpticStudio.
- Surface scattering is not currently supported on surfaces inside a black box.

The encryption is based upon a 256-bit algorithm that provides good, but not unbreakable security. This feature is provided as is and the user should determine if the encryption implemented provides adequate security for the intended application before distributing sensitive data in the encrypted format. Zemax LLC provides no warranty and assumes no liability for the use of this feature.

Encrypted Coating

This feature exports a single coating to a Zemax Encrypted Coating (ZEC) file
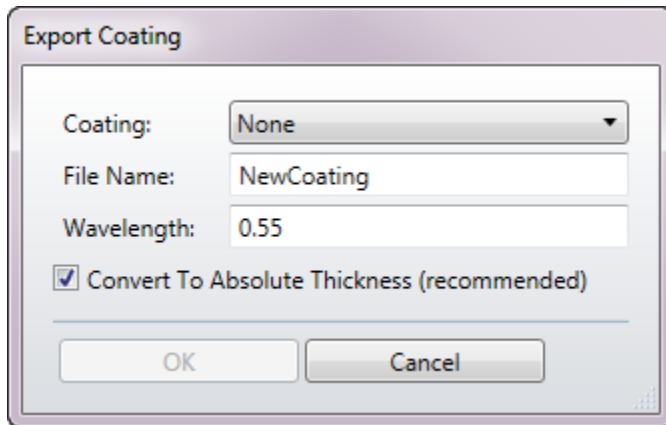
Coating: The name of the coating to export.
File Name: The name of the file to create. The file name should not include any path or extension. The file will be created in the current coating file folder and the extension "ZEC" will be automatically appended.
Wavelength: The wavelength, in micrometers, measured in air at system temperature and pressure, used to define the layer thicknesses. This value is only used when converting layers from optical thickness to absolute thickness.
Convert to Absolute Thickness: Converts optical thickness defined relative to the primary wavelength (for example, a quarter wave thickness) in the coating to absolute thickness. For maximum security, this option should be selected. If the relative optical thickness is left in the coating definition, then the end user of the encrypted coating may be able to determine some properties of the coating by changing the design wavelength. The conversion from thickness in primary waves to absolute thickness is done using the wavelength value from the above setting.

General Points to Note

Optical coating designers and manufacturers are often hesitant to divulge the details of their coating designs. This feature allows a coating defined in Zemax format to be exported to a single file that is encrypted. The encrypted file may be

distributed to users that need to trace rays and perform analysis on the coating as applied to a specific optical system, without the details of the coating design being revealed.

The coating data that is exported includes all the material, index, extinction, dispersion, and layer data required to exactly reproduce the coating performance. No approximations or tables are used. However, no coating tapers, ideal coatings, or table coatings are supported. A maximum of 12 different materials and 400 layers may be used in defining the coating to be exported.

To use an encrypted coating file, see "The ENCRYPTED data section" in the Coating File description.
.
The encryption is based upon a 256-bit algorithm that provides good, but not unbreakable security. This feature is provided as is and the user should determine if the encryption implemented provides adequate security for the intended application before distributing sensitive data in the encrypted format. Zemax LLC provides no warranty and assumes no liability for the use of this feature.

Convert Group
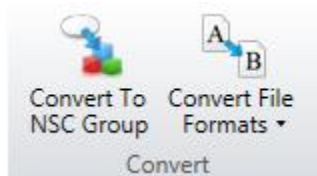
This group gives access to the Convert to NSC tool and various file conversion utilities

Convert to NSC Group

Converts a range of surfaces in the Lens Data Editor into a group of 3D objects in either a Non-Sequential Components surface or a new non-sequential system.

It is recommended that you insert dummy surfaces before and after the range of surfaces you wish to convert, and then choose these as the first and last surfaces for theconversion
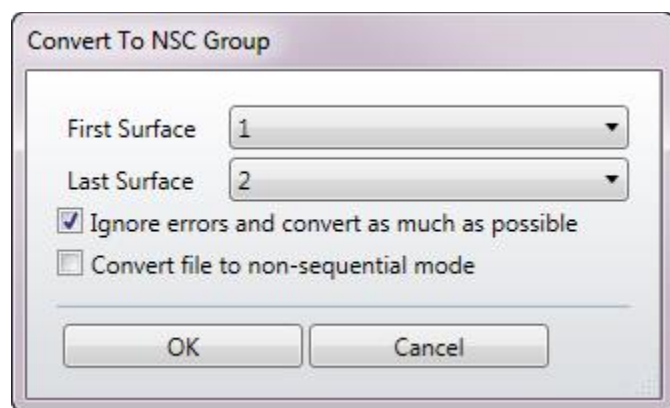
First Surface: This is the first surface of the lens group to be converted to non-sequential components.
Last Surface: The last surface of the lens group to be converted.

Ignore errors and convert as much as possible: If checked, the conversion will not stop even if errors or limitations in the conversion process are detected. This switch is useful because often the majority of a system may be converted properly even though a small portion of the system cannot be converted.

Convert file to non - sequential mode: If checked, the selected range of surfaces will be converted into a "pure" non-sequential file. In this case, the field positions, pupil data, and stop location are not considered.

General Points to Note

This feature converts a range of sequential surfaces in the Lens Data Editor into a group of non-sequential components. **Don't assume this feature perfectly converts your sequential surface data; carefully check the conversion results before doing any important analysis.**

There are some restrictions on the ability of OpticStudio to convert sequential surfaces to NSC objects:

- The surfaces must all be of type Standard, Even Aspheric, Toroidal, Biconic, or Coordinate Break. This restriction will be relaxed as more NSC object types and more conversion logic is added to OpticStudio.
- All coordinate break surfaces must be between elements, in "air" and not embedded within the lenses. This is the normal condition of use.
- The surface prior to the first surface and the last surface in the selected range must both be made of "air" with unity index.
- The stop must precede the first surface. **This restriction is removed** if the "Convert file to non-sequential mode" box is checked.
- OpticStudio can only convert rectangular, circular, and elliptical apertures on standard surface mirrors. Most other combinations of non-circular apertures and surface types may not convert properly. Non-circular apertures on standard surfaces are modeled using user defined aperture (UDA) files that OpticStudio automatically creates. For more information on UDA's see "User defined apertures and obscurations" on page 82.

There are times when the ray tracing after conversion will not yield exactly the same results as before. The possible explanations for this include the following:

- Some rays may not trace the same inside a non-sequential group if some of the surfaces have central obscurations or annular apertures. This is especially likely to occur in telescopes with obscured apertures. In the non-sequential representation, the chief ray will either not make it through the group or may take a radically different path.
- Rays that are exactly at the edge of a sequential surface may miss the surface, or strike the outer cylinder wall of a lens, in the non-sequential trace. For this reason, OpticStudio adds a very small increase in the diameter of each surface when converting to a NSC group.
- For systems with concave surfaces at the ends of the selected range, the edges of the surface may fall outside of the entry and/or exit ports, which will cause ray tracing errors. The solution is to manually edit the entry/exit port locations, or, insert dummy surfaces before and after the selected range prior to conversion.

The feature works by the following algorithm:

- First a Non-Sequential Component surface and a dummy exit port surface are inserted. The exit port is positioned to coincide with the last surface in the defined range.
- The range of surfaces is converted into either NSC surface objects (for mirrors) or NSC lens objects (for pairs of surfaces with glass in between).
- Coordinate breaks are replaced with null objects with equivalent rotations; and subsequent objects will reference the null objects.
- The original surfaces are then deleted.
- If the "Convert file to non-sequential mode" box is checked, then the resulting NSC surface is the only surface retained and the file is placed in NSC mode.

The feature may not work correctly in all cases. Some additional dummy surfaces with small spacings may need to be inserted prior to and after the range of lens surfaces to be converted. The feature may also not work for lenses with multi-configuration data that alters the thicknesses or glass types of any surfaces within the range of surfaces to convert.

Convert File Formats

This feature is used to convert data files from other programs into the equivalent Zemax format. Or to take Zemax data into the other program's format.
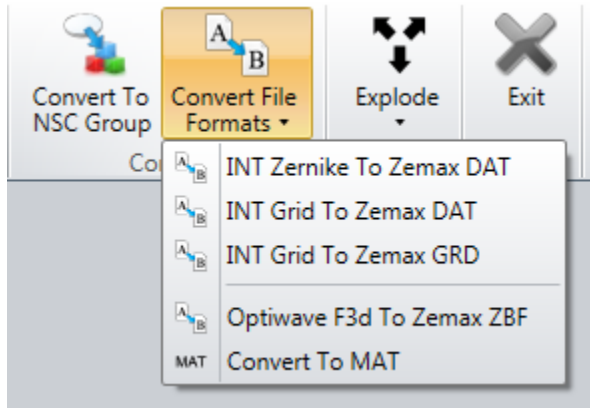
INT Zernike to Zemax .DAT

The INT file format is widely used by interferometer manufacturers as a data export format. After conversion these files can be read by the Import Tool on the Lens Data Editor toolbar into Zernike Fringe or Zernike Sag Surfaces
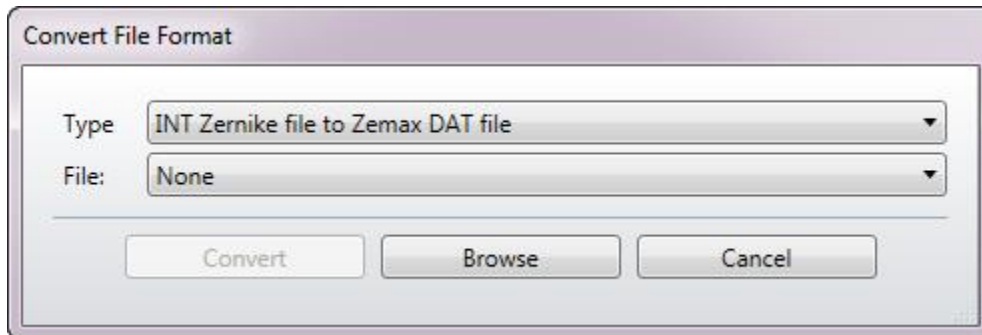
File: The file name to be converted. Only files in the most recently read folder with the proper extension are listed in the drop-down box.
Browse: Select a file from any other folder.
Convert: Start the conversion. Any erorr messages on conversion will be generated during this process.

The file format for .INT Zernike files is as follows:

```
! comment (optional)
! comment (optional)
Title line. A title line is required.
<zerntype> <numterms> <datatype>
<value of Zernike 1 term>
<value of Zernike 2 term>
```

There may be any number of comment lines at the beginning of the file. All comment lines must start with the "!" symbol. The first line in the file that does not start with the "!" symbol is the title line. All .INT files must have a title

line (which can consist of up to 80 characters). The next line in the file consists of various parameters which define the format of the file.

<zerntype> defines the Zernike format (either "ZRN" for Zernike Standard or "ZFR" for Zernike Fringe). **At this time, only the Zernike Fringe format (ZFR) is supported**.

<numterms> specifies the number of Zernike terms that will be provided in the file.

<datatype> refers to what type of data is stored in the file (either "SUR" for surface deformation data, "WFR" for wavefront deformation data or "FIL" for intensity apodization filter data). Zemax does not differentiate between the three different types of data when importing the data to a Zernike surface.

The remaining lines in the file contain the Zernike coefficients of the corresponding polynomial, either Zernike Standard or Zernike Fringe.

INT Grid to Zemax .DAT

The INT file format is widely used by interferometer manufacturers as a data export format. After conversion these files can be read by the Import Tool on the Lens Data Editor toolbar into Grid Sag or Grid Phase sequential surfaces.

File: The file name to be converted. Only files in the most recently read folder with the proper extension are listed in the drop-down box.

Browse: Select a file from any other folder.

Convert: Start the conversion. Any erorr messages on conversion will be generated during this process.

The file format for .INT Grid files is as follows:

```
! comment (optional)
! comment (optional)
Title line. A title line is required.
GRD <xsize> <ysize> <datatype> WVL <wavelength> SSZ <scalesize> NDA
<nodatavalue>
XSC <xscale>
<data value 1> <data value 2> <data value 3> . . . . . . . . . <data value n>
<data value n+1> . . .
```

There may be any number of comment lines at the beginning of the file. All comment lines must start with the "!" symbol. The first line in the file that does not start with the "!" symbol is the title line. All .INT files must have a title line (which can consist of up to 80 characters). The next line in the file consists of various parameters which define the format of the file. For Grid Files, this line must start with a "GRD" statement.

<xsize> specifies the number of grid points in the x-direction while <ysize> refers to the number of grid points in the y-direction.

<datatype> refers to what type of data is stored in the file (either "SUR" for surface deformation data, "WFR" for

wavefront deformation data or "FIL" for intensity apodization filter data). Zemax does not differentiate between the three different types of data, the interpretation is determined by whether the data is imported to a Grid Sag or Grid Phase surface.

<wavelength> is the wavelength, in microns, at which the data was measured. The <wavelength> value must be prefaced a "WVL" statement.

<scalesize> specifies the value for the input data values which corresponds to one wave of deformation. The <scalesize> value must be prefaced by an "SSZ" statement.

<nodatavalue> specifies the value of the input data that should be interpreted as missing data. Any rays that are incident in these regions of the grid are vignetted. This is an optional parameter. If provided, the <nodatavalue> parameter must be prefaced by an "NDA" statement.

 <xscale> is the ratio of the size of the grid in the x-direction to the size of the grid in the y-direction. This is an optional parameter. If the value is omitted, a value of 1.0 (which corresponds to a square grid) is used. If provided, the <xscale> parameter must be prefaced by an "XSC" statement.

The remaining lines in the file contain the individual grid point values. There should be a total of <xsize> * <ysize> values specified. Note that this data is in "free format". As such, each line of data does not necessarily correspond to a single row of data on the actual grid. The data is specified from -x to +x starting at the +y edge of the grid.

INT Grid to Zemax .GRD

The INT file format is widely used by interferometer manufacturers as a data export format. After conversion these files can be read into the Grid Sag non-sequewntial objecy

File: The file name to be converted. Only files in the most recently read folder with the proper extension are listed in the drop-down box.

Browse: Select a file from any other folder.

Convert: Start the conversion. Any erorr messages on conversion will be generated during this process.

The file format for .INT Grid files is as follows:

```
! comment (optional)
! comment (optional)
Title line. A title line is required.
GRD <xsize> <ysize> <datatype> WVL <wavelength> SSZ <scalesize> NDA
<nodatavalue>
XSC <xscale>
<data value 1> <data value 2> <data value 3> . . . . . . . . . <data value n>
<data value n+1> . . .
```

There may be any number of comment lines at the beginning of the file. All comment lines must start with the "!" symbol. The first line in the file that does not start with the "!" symbol is the title line. All .INT files must have a title

line (which can consist of up to 80 characters). The next line in the file consists of various parameters which define the format of the file. For Grid Files, this line must start with a "GRD" statement.

<xsize> specifies the number of grid points in the x-direction while <ysize> refers to the number of grid points in the y-direction.

<datatype> refers to what type of data is stored in the file (either "SUR" for surface deformation data, "WFR" for wavefront deformation data or "FIL" for intensity apodization filter data). Zemax does not differentiate between the three different types of data, the interpretation is determined by whether the data is imported to a Grid Sag or Grid Phase surface.

<wavelength> is the wavelength, in microns, at which the data was measured. The <wavelength> value must be prefaced a "WVL" statement.

<scalesize> specifies the value for the input data values which corresponds to one wave of deformation. The <scalesize> value must be prefaced by an "SSZ" statement.

<nodatavalue> specifies the value of the input data that should be interpreted as missing data. Any rays that are incident in these regions of the grid are vignetted. This is an optional parameter. If provided, the <nodatavalue> parameter must be prefaced by an "NDA" statement.

<xscale> is the ratio of the size of the grid in the x-direction to the size of the grid in the y-direction. This is an optional parameter. If the value is omitted, a value of 1.0 (which corresponds to a square grid) is used. If provided, the <xscale> parameter must be prefaced by an "XSC" statement.

The remaining lines in the file contain the individual grid point values. There should be a total of <xsize> * <ysize> values specified. Note that this data is in "free format". As such, each line of data does not necessarily correspond to a single row of data on the actual grid. The data is specified from -x to +x starting at the +y edge of the grid.

OptiWave® f3d Beam File to Zemax ZBF

OptiWave F3D files are used to describe either linear polarized or generally polarized electric fields. These files are converted to the Zemax Beam File (ZBF) format used by the Physical Optics Propagation and Partially Coherent Image Analysis features.
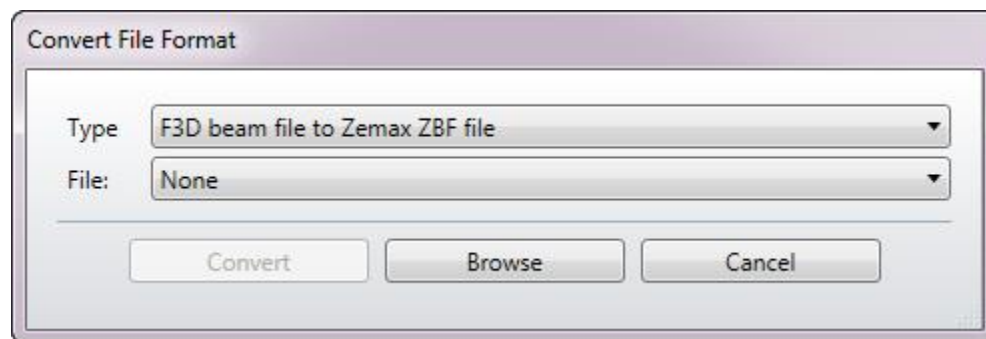
File: The file name to be converted. Only files in the most recently read folder with the proper extension are listed in the drop-down box.
Browse: Select a file from any other folder.
Convert: Start the conversion. Any erorr messages on conversion will be generated during this process.

The F3D file format consists of 3 lines of header information of the form

```
BCF3DCX 3.0
Nx Ny
xmin xmax ymin ymax
```

The identifier BCF3DCX identifies a linear polarized beam file. Only file format version 3.0 is currently supported.

The values Nx and Ny are the integer number of points in each direction. Zemax ZBF files only support powers of 2, while F3D files may have any integer number of points. If Nx or Ny are not powers of two, the ZBF representation will be zero-padded to make the beam dimensions equal to the nearest larger power of 2. For example, if there are 115 points in the F3D file, the ZBF will contain 128 points, with zero values added to center the original data.

The values for xmin, xmax, ymin, and ymax define the physical size of the beam. All these values must all be in units of micrometers. Zemax will scale these values to millimeters, and use millimeters for the exported beam size. ZBF files only contain the x- and y-direction widths of the beam, and not any absolute position.

The F3D format does not include any wavelength data, and all E-Field phase data is referenced to a local plane. Zemax will fit a Gaussian beam to the E-Field data to estimate the pilot beam properties required for subsequent propagation in Zemax, and store this data in the ZBF file

Convert to MAT

This feature reads the data from a ZRD file or a ZBF file then writes it to the native Matlab® data file format, the MAT file. **This feature requires the presence of a Matlab installation to run** but does not require Matlab to run concurrently with Zemax.
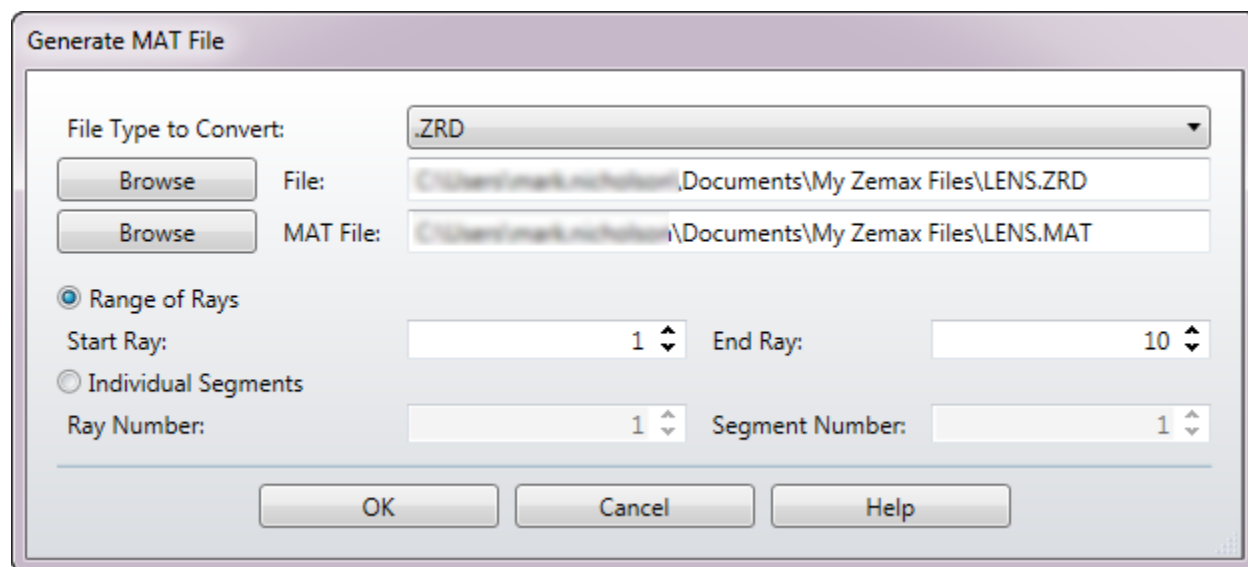
File Type to Convert: Selects the ZRD or ZBF file type conversion.
File: The file name and path of the ZBF or ZRD file to be converted.
MAT File: The file name and path of the MAT file to be created.
Range of Rays If a ZRD file is to be converted this button tells Zemax that a range of rays from the ZRD file will be written to the MAT file.
Start Ray: The first ray to be converted, the Ray Database viewer will give you the number.
End Ray: The last ray to be converted, the Ray Database viewer will give you the number.
Individual Segments: If a ZRD file is to be converted this button tells Zemax that a specific segment from the ZRD file will be written to the MAT file.
Ray Number: The ray from which the segment is to be converted, the Ray Database viewer will give you the number.
Segment Number: The number of the segment from the selected ray the Ray Database viewer will give you the number.
Range of Surfaces: If check a range of surfaces associated with a ZBF will be converted.
Start Surface: The first surface to be converted, the value of which must be greater than zero.
End Surface: The last surface to be converted, the value of which must be greater than zero.

The ray and segment numbers are displayed in the Ray Database Viewer.

The **Range of Surfaces** option requires you to have created ZBF files over the surfaces specified by the Start Surface and End Surface. When the Range of Surfaces option is checked the file name format of the ZBF files to be converted should be filename_000x.ZBF, where x is a positive integer. When this option is used Zemax will add the string _000x to the ZBF filename specified in the File box. For example, if the File box content is "C:\Path\filename.ZBF" then before reading the ZBF file for conversion Zemax will modify this to "C:\Path\file- name_000x.ZBF", where x will be the inclusive range of integers specified in the Start Surface and End Surface box. OpticStudio then cycles over these files and generates a Matlab data structure with the contents of the files. If this option is not checked then the file with a name and path as defined in File box will be used.

If you receive an error message indicating that there are DLLs missing please check that they are present in the bin\win32 or bin\win64 folders of your Matlab installation. Also check that the Folders tab of the Preferences dialog points to the folder containing your Matlab installation. It should be noted that the 64-bit version of OpticStudio will require a 64-bit version of Matlab and the 32-bit version of OpticStudio will require a 32-bit version of Matlab.

Explode

Explode is used to explode CAD assemblies in various formats into individual parts using the ZOF (Zemax Object Format) structure.
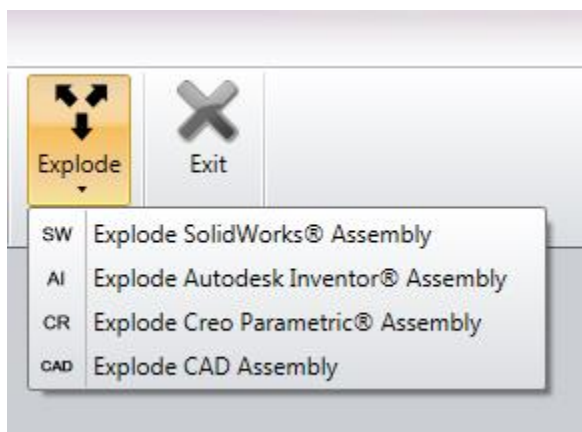
CAD Assemblies include STEP, IGES and STL files

The Setup Tab

This tab provides access to OpticStudio's Setup features.

This is the Setup tab in Sequential UI Mode:
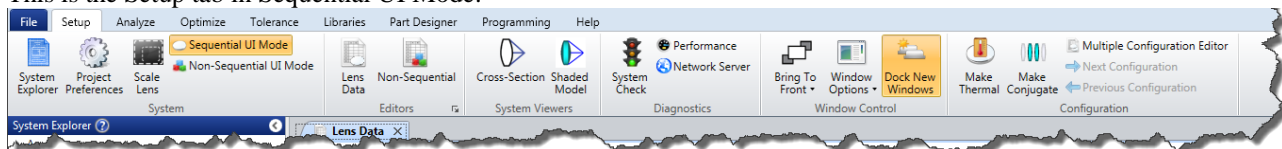
This is the Setup tab in Non-Sequential UI Mode:
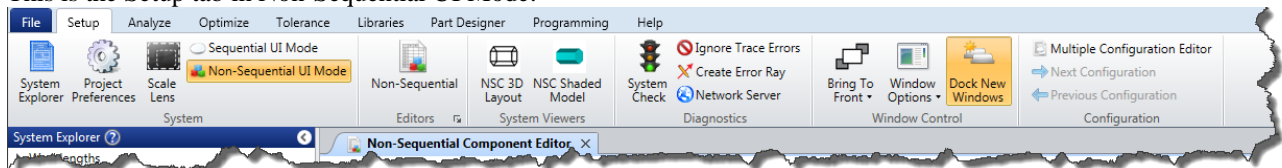
System Group

These are the System Setup tools.
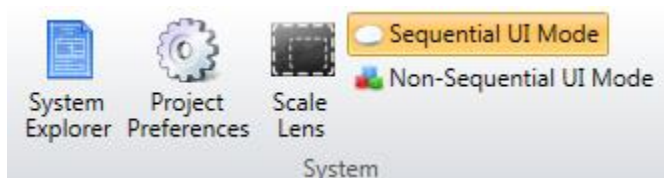
System Explorer

The System Explorer button is available in the System section of the Setup tab. This arrow button expands the System Explorer sidebar, which is used to define data that pertains to the lens as a whole system, rather than data associated with a single surface. The data in the System Explorer contains most of the basic optical and mechanical definitions used in your Indigo project.

This is the System Explorer in Sequential UI Mode:
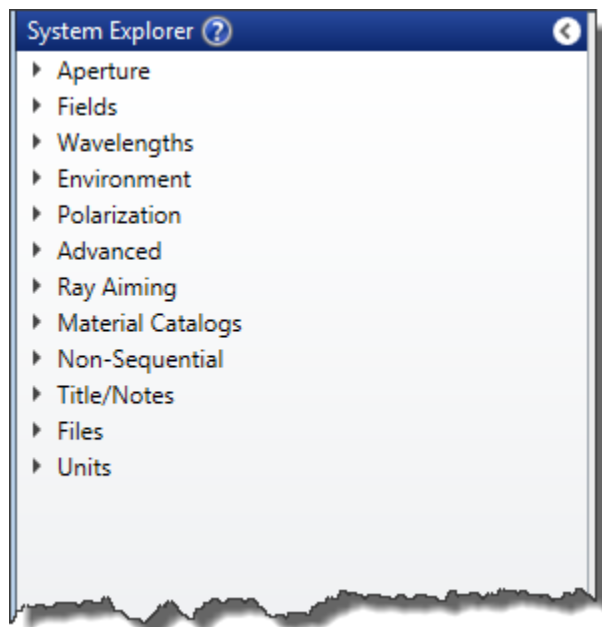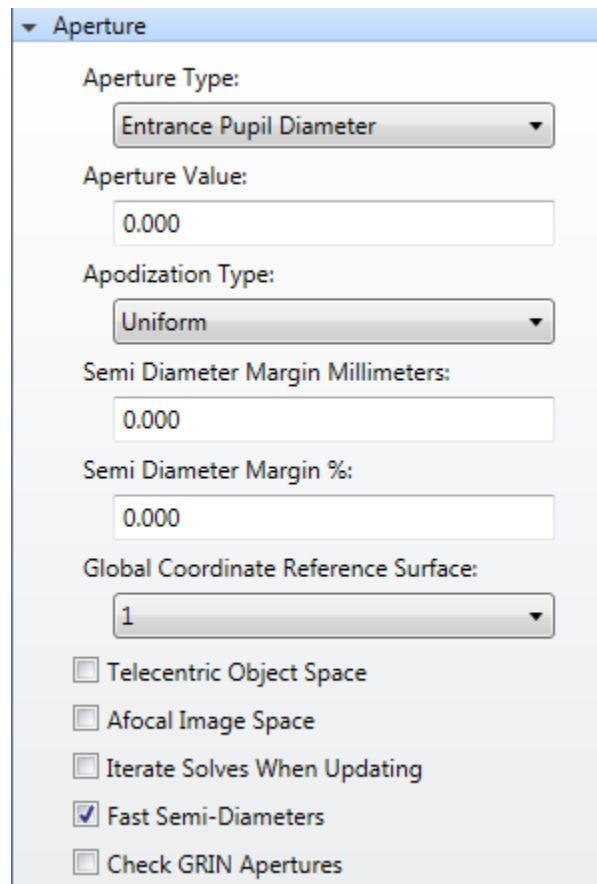
This is the System Explorer in Non-Sequential UI Mode:
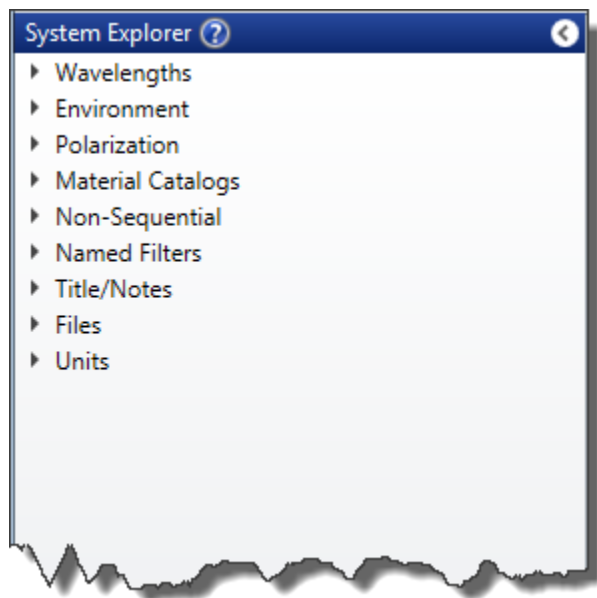
See the topics below for more information.

Aperture

These Aperture features are available in the System Explorer. The system aperture defines the size of the beam through the system on axis. To set the system aperture, you need to define the system aperture type and the system aperture value.

Aperture Type

Entrance Pupil Diameter The diameter of the pupil in lens units as seen from object space.

Image Space F/# The infinite conjugate paraxial F/# in image space.

Object Space Numerical Aperture The numerical aperture ($n \sin \theta_m$) of the marginal ray in object space.

Float By Stop Size Defined by the semi-diameter of the stop surface.

Paraxial Working F The paraxial F/# in image space for the defined conjugates.

Object Cone Angle The half angle in degrees of the marginal ray in object space, which may exceed 90 degrees. Object cone angle may not be used if the entrance pupil is virtual, that is, if the distance from the object to the entrance pupil is negative. When using object cone angle, the default "Uniform" apodization of rays in the pupil is uniform in angle space, rather than uniform on a plane. If the apodization type is set to "Cosine Cubed" then the rays are uniformly distributed in solid angle, as would be correct for a point source radiating uniformly in all directions. See "Apodization Type". This ray distribution may be significantly different than other aperture type settings for large cone angles.

These terms are further defined in "Conventions and Definitions". If you select "Object Space NA" or "Object cone angle" for a system aperture type, the object thickness must be less than infinity. Only one type of system aperture may

be defined per configuration; for example, once the Entrance Pupil Diameter is specified, all the other aperture definitions are determined by the lens data prescription.

Aperture Value

The system aperture value meaning depends upon the system aperture type selected. For example, if "Entrance Pupil Diameter" is selected as the system aperture type, the system aperture value is the entrance pupil diameter in lens units. ZEMAX uses the system aperture type and system aperture value together to determine such fundamental quantities as the entrance pupil size, and clear apertures for all components.

Selecting "Float by Stop Size" for a system aperture type is the only exception to this rule. The semi-diameter of the stop surface (set on the Lens Data Editor) is used to define the system aperture if "Float by Stop Size" is selected for a system aperture type.

Apodization Type

Uniform Rays are distributed uniformly over the entrance pupil, simulating uniform illumination. By default, the pupil is always illuminated uniformly.

Gaussian Imparts an amplitude variation over the pupil that is Gaussian in form. The apodization factor refers to the rate of decrease of the beam amplitude as a function of radial pupil coordinate. The beam amplitude is normalized to unity at the center of the pupil. The amplitude at other points in the entrance pupil is given by

$$A(\rho) = e^{-G\rho^2}$$

where G is the apodization factor and ρ is the normalized pupil coordinate. If the apodization factor is zero, then the pupil illumination is uniform. If the apodization factor is 1.0, then the beam amplitude has fallen to the 1 over e point at the edge of the entrance pupil (which means the intensity has fallen to the 1 over e squared point, about 13% of the peak). The apodization factor can be any number greater than or equal to 0.0. Values larger than about 4.0 are not recommended. This is because most computations will sample too few rays to produce meaningful results if the beam amplitude falls too quickly at the edges of the pupil.

Cosine cubed Simulates the intensity fall off characteristic of a point source illuminating a flat plane. Note cosine cubed apodization is only useful and should only be used for point sources or field points close to the axis when compared to the entrance pupil diameter. For a point source, the intensity of a ray illuminating a differential area on a plane is given by

$$I(\theta) = (\cos\theta)^3$$

where θ is the angle between the z axis and the ray intersecting the entrance pupil, and the relative intensity at the center of the pupil is 1.0. Converting to normalized pupil coordinates and taking the square root yields the pupil coordinate amplitude apodization

$$A(\rho) = \frac{1}{(1 + (\rho \tan\alpha)^2)^{\frac{3}{4}}}$$

Where $\tan\alpha$ is the tangent of the angle between the z axis and the marginal ray. Zemax uses the entrance pupil position and size to compute $\tan\alpha$. The apodization factor is not used by the cosine cubed apodization.

If the aperture type is "Object Cone Angle" then a slightly different apodization technique is used. The apodization is achieved by careful selection of the ray cosines in object space. The resulting distribution is uniform in solid angle rather than angle space. The solid angle Ω of a sphere illuminated by a cone of angle θ is given by

$$\Omega = 2\pi(1 - \cos\theta) .$$

The normalized radial pupil coordinate *p* is related to the fractional solid angle by

$$p^2 = \frac{1 - \cos\theta}{1 - \cos\alpha}$$

where α is the maximum cone angle defined by the system aperture value and θ is the angle of the ray with the normalized pupil coordinate *p*. This apodization will yield uniform ray density and power per solid angle everywhere over the defined cone.

User Defined Zemax also supports user defined apodizations on any surface, rather than just the entrance pupil. User defined surface apodizations are implemented using the user defined surface type described in "User defined surface apodization using DLLs".

Apodization Factor

The apodization factor determines how fast the amplitude decays in the pupil. Used only for Gaussian apodization. See the previous section on the Apodization Type for details. More information on apodization can be found in the following Knowledge Base article: What Does the Term 'Apodization' Mean?

Semi Diameter Margin

The semi-diameter of every surface in "automatic" mode is computed to be the radial aperture required to pass all rays without clipping. For systems with closely spaced elements in or near edge contact, this yields surface apertures that provide no clearance for finishing or mounting. Often, optical surfaces are only well finished within some fraction of the full radial aperture, typically between 90% and 98%, depending upon the part size.

Margin Millimeters This semi diameter margin control allows specification of an additional amount of radial aperture as a fixed number. The default value of zero leaves no margin.

Margin % This semi diameter margin control allows specification of an additional amount of radial aperture as a percentage. The default value of zero leaves no margin, while a margin of 5% would add 5% to the semi-diameters of all surfaces under "automatic" control. The maximum allowed margin is 50%. The semi-diameter margin is not added to the stop surface.

If both "percent" and "millimeters" margin values are non-zero, the percent is added first, then the lens unit margin.

Global Coordinate Reference Surface

Global coordinates are defined by a rotation and translation from the local coordinates on each surface. The rotation matrix and the offset vector can be computed for any surface using any other surface as a global reference. The default reference surface is 1, but any other surface may be chosen. The only exception occurs when the object is at infinity, and then 0 may not be the reference surface. The surface selected determines the global coordinate system origin location and orientation. The reference surface is also used to define the point of overlap for multiple zoom positions on the 3D layout plot.

Telecentric Object Space

If this checkbox is selected, Zemax will assume the entrance pupil is located at infinity, regardless of the location of the stop surface. All chief rays leaving the object surface will be parallel to the local Z axis. Ray aiming and field points defined using angles may not be used when this option is selected. For best performance, set the stop surface to surface 1 when using telecentric mode.

Afocal Image Space

If this box is checked, Zemax will perform most analysis features in a manner appropriate for optical systems with output beams in image space that are nominally collimated. Strictly speaking, the term afocal means the optical beam is

collimated in both object and image space. However, Zemax uses this term to describe collimated output in image space, whether the beam is collimated in object space or not. For brevity, the documentation uses the term "afocal mode" to indicate Afocal Image Space is checked on, and "focal mode" to indicate Afocal Image Space is checked off.

When using afocal mode, transverse, longitudinal, and MTF aberrations are computed in units appropriate to afocal systems. Transverse aberrations are computed as a function of angle relative to a reference ray rather than units of length. Longitudinal aberrations are computed as a function of defocus in diopters (inverse meters) rather than as defocus in units of length. MTF is measured in cycles per angle rather than cycles per length. The change of units is not always mentioned in the documentation for each individual analysis feature. When using afocal mode, the user should be aware that some analysis window settings described in focal units are actually in afocal units if afocal mode is on. Afocal aberrations may be measured in various units, with milliradians being the default. To change the afocal unit type, see "Afocal Mode Units" in the Units section of the System Explorer.

Wavefront error (or Optical Path Difference) is measured somewhat differently in afocal mode than in focal mode. For afocal systems, the OPD is measured relative to a reference plane that is normal to the chief ray of the reference wavelength. If the OPD is referenced to the exit pupil (the recommended setting) then the afocal OPD is measured at the plane located at the exit pupil position relative to the image surface. If the OPD reference is set to absolute then the OPD is referenced to a plane located at the image surface. Note that a shift in the location of the plane at which the afocal OPD is referenced will in general change the OPD values, unless the beam is perfectly collimated. For more information on the OPD reference, see "Reference OPD" in the Advanced section of the System Explorer.

Analysis features which compute data as a function of defocus use units of length in focal mode and diopters in afocal mode. Note that a positive defocus in diopters means the image shifts toward the lens, which is a negative shift in length. For this reason through focus plots will appear to flip left-to-right when switching between a focal mode system and an afocal mode system with the same fundamental aberrations.

Other than the change of units, most Zemax features work exactly the same in focal and afocal mode. Some features do not have meaning for afocal systems and an error message or possibly meaningless data will result if the analysis is attempted in afocal mode. To optimize afocal systems, use one of the angle based default merit functions described in "Selecting the type of optimization".

Iterate Solves When Updating

Solves placed on parameters in the Lens Data Editor sometimes require iteration to compute accurately. Typically this includes curvature and thickness solves that depend upon ray tracing. The tracing of specific rays, such as the marginal or chief ray, depends upon pupil positions. These pupil positions may change when the solve value is updated, creating the need for iteration. The primary symptom of the need for iteration is when the merit function increases or jumps around erratically during optimization.

Zemax normally detects the need for iteration. However, if Zemax does not iterate automatically, checking this option will force iteration on all solves when updating the lens data. Iteration does slow down calculations, particularly optimization, and so this option should only be checked on when required.

Fast Semi-Diameters

Zemax computes "automatic" semi-diameters to estimate the clear aperture required on each surface to pass all rays at all field points and wavelengths. For axial systems, this computation can be done accurately by tracing just two rays, the top and bottom marginal rays, for each field and wavelength. By default, for axial systems Zemax will trace just these two rays in the "true" tangential plane (for a definition of this term see "Sagittal and Tangential") of the vignetted pupil for each field and wavelength, then use the radial coordinates of each ray at each surface to determine the semi-diameter required.

For some non-axial systems, the resulting estimate is not accurate enough. This typically includes systems which have tight edge and clear aperture constraints. For these systems, there is no general way of accurately computing the semi-diameters other than by tracing a large number of rays in the vignetted pupil. Zemax will iteratively trace as many

marginal rays around the perimeter of the pupil as required to determine every semidiameter to an accuracy of about 0.01% (5 significant figures).

Iteration can be slow, since Zemax needs to update the semi-diameters frequently, especially during optimization. There is a trade-off between speed and accuracy. For axial systems, Zemax will use the slower iterative algorithm only if the "Fast Semi-Diameters" option is checked "off"

Selected (On) For non-axial systems, Zemax will automatically use the slower iterative algorithm. If the "Fast Semi-Diameters" option is checked "on", Zemax will trace only as many marginal rays as required to estimate the automatic semi-diameters to about 0.01%. The algorithm starts by tracing 2, then 4, then 8 rays, and so on, until the semi-diameter values converge to within 0.01% of the previous estimate.

Not Selected (Off) For non-axial systems, if the "Fast Semi-Diameters" option is checked "off", Zemax will trace at least 32 rays around the vignetted pupil at each field and wavelength, and more if required to estimate the automatic semi-diameters to about 0.01%. Tracing at least 32 rays is safer, but slower, and should only be used when very high precision is required in the automatic semi-diameter calculations.

Both methods used by Zemax to determine semi-diameters are accurate only for surfaces whose maximum size is set by the radial coordinate of a marginal ray. The methods do not work for surfaces that lie in a caustic. For surfaces in a caustic, the exact ray-based semi-diameter could be determined using a spot diagram or footprint analysis, however this level of precision is likely never required since diffraction effects, which the rays do not model, would be significant.

Check GRIN Apertures

If checked, this setting instructs Zemax to check all gradient index ray traces for surface aperture vignetting. Each GRIN trace within the media is checked to see if the ray ever passes outside the front surface aperture boundary, and if it does, then the ray is vignetted. If this setting is not checked, then the ray may travel outside the boundary defined on the front surface, as long as the ray passes the aperture at the surface.

Fields

These field settings are available in the System Explorer. This feature is used to define the number, type, and magnitude of the field points to be used for tracing rays. Vignetting factors are also set in this dialog box.

Field Types:

Angle Field angles are always in degrees. The angles are measured with respect to the object space z axis and the paraxial entrance pupil position on the object space z axis. Positive field angles imply positive slope for the ray in that direction, and thus refer to negative coordinates on distant objects.

Object Height Measured in lens units.

Paraxial Image Measured in lens units. When paraxial image heights are used as the field definition, the heights are the paraxial image coordinates of the primary wavelength chief ray on the paraxial image surface, and if the optical system has distortion, then the real chief rays will be at different locations.

Real Image Measured in lens units. When real image heights are used as the field definition, the heights are the real ray coordinates of the primary wavelength chief ray on the image surface.

Normalization:

Radial Height If the field normalization is radial, then the normalized field coordinates represent points on a unit circle. The radius of this unit circle, called the maximum radial field, is given by the radius of the field point farthest from the origin in field coordinates. The maximum radial field is then used to scale all fields to normalized field coordinates.

Rectangular If the field normalization is rectangular, then the normalized field coordinates represent points on a unit rectangle. The x and y direction widths of this unit rectangle, called the maximum x field and maximum y field, are defined by the largest absolute magnitude of all the x and y field coordinates. The maximum x and y field is then used to scale all fields to normalized field coordinates.

Vignetting:

Set Vignetting Recomputes the vignetting factors for each field based upon the current lens data. Vignetting factors (VDX, VDY, VCX, VCY, VAN) are coefficients which describe the apparent entrance pupil size and location for different field positions. These vignetting factors should be left at zero if there is no vignetting in the system. The factors are further described in "Conventions and Definitions" under "Vignetting factors". The set vignetting algorithm estimates the vignetting decenter and compression factors so that the four marginal rays in the top, bottom, left, and right edges of the pupil pass within the apertures of each surface. Only the primary wavelength is used.

Clear Vignetting Clears the vignetting factors back to the default values of zero.

The algorithm starts by launching a grid of rays through the pupil. At each surface with a surface aperture, the ray is tested to see if it passes within the specified aperture. All rays which pass all surfaces are then used to compute the unvignetted pupil centroid. The edge of the unvignetted pupil is then computed using an iterative method accurate to about 0.001%.

The algorithm may not work in all cases. For systems where the set vignetting algorithm fails, the vignetting factors will need to be adjusted manually. The accuracy of the set vignetting algorithm can be tested by tracing a few marginal rays.

See "Vignetting Factors" under "Conventions and Definitions" .

Field Data Editor

The Field Data editor is accessed by double clicking the "Fields" bar in the System Explorer.
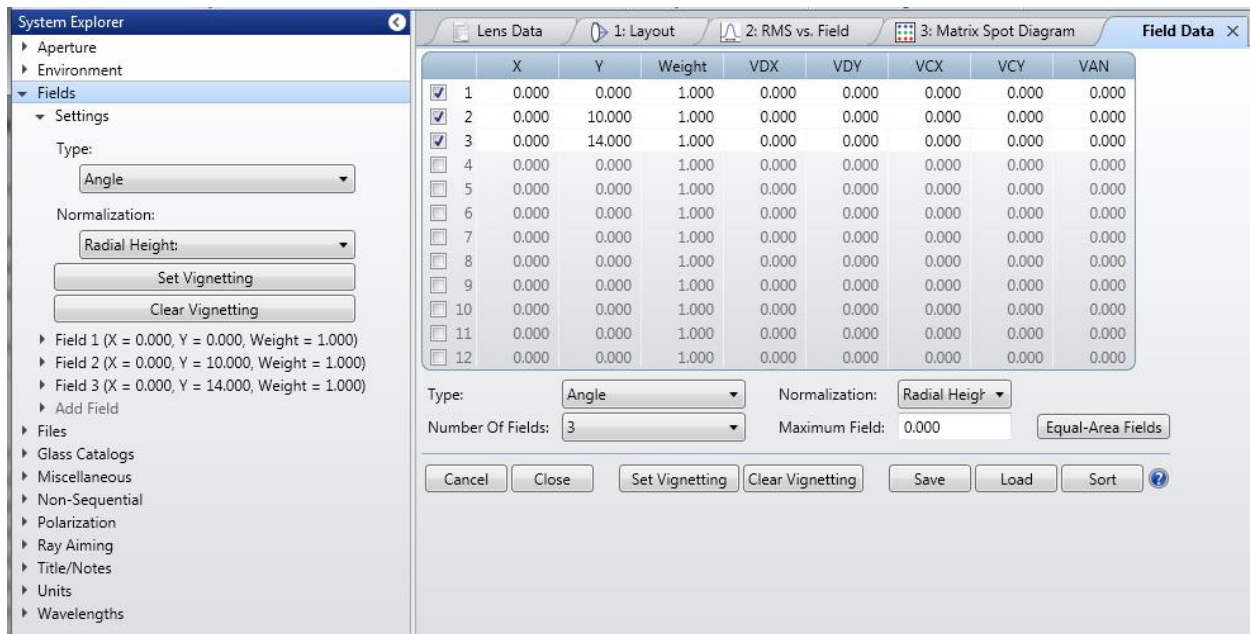
The Field Data editor includes the same information and settings as the Fields section of the System Explorer, but also includes a button for automatically defining equal area fields. Select the number of fields and the maximum field value, and Zemax will populate the dialog with field points that divide the field of view into regions of equal area. The Save and Load buttons are used to store and retrieve the field data independently from the lens data. The file format is text that can be edited or created outside of Zemax.

Defining wide-angle fields

Zemax assumes that the on-axis (0,0) object space field lies within the field of view of the optical system. The recommended setup is for this field to define the center of the field of view. Sometimes optical systems may be defined centered on an off-axis field for various reasons. Note, however that if the (0,0) object space field cannot trace through the system, raytrace errors can occur for all fields. In these cases Zemax will issue an error in the System Check report indicating that the "gut ray failed". If this occurs, the solution is to redefine the system so that the (0,0) field lies close to the center, or at least within the field of view.

Wavelengths

The Wavelength settings are available in the System Explorer.
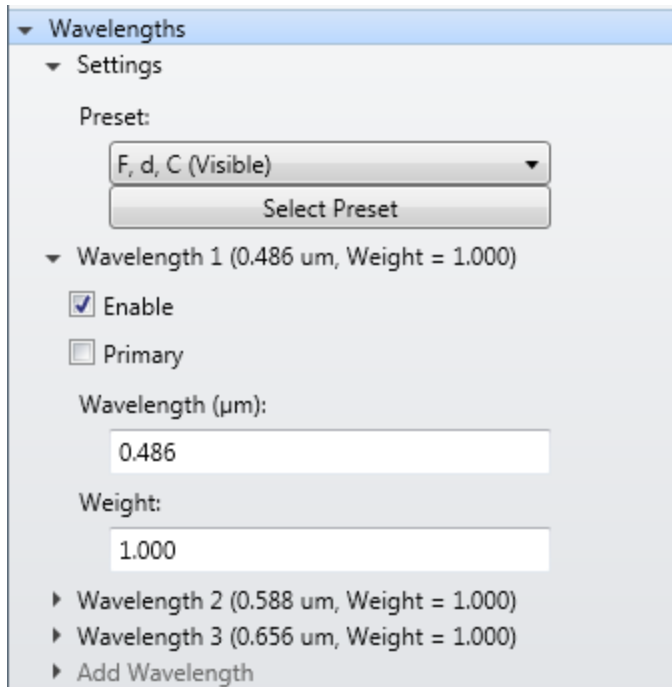
Settings:

The wavelengths dialog box is used to set wavelengths, weights, and the primary wavelength. The buttons can be used to activate and deactivate wavelengths, and to sort, save, and load the data. A list of commonly used wavelengths is also included. To use the entries on the list, select the desired wavelength set, and click on the "Select Preset" button.

Wavelength data are always measured in micrometers referenced to "air" at the current system temperature and pressure. The default system temperature is 20 degrees Celsius, and the default air pressure is 1.0 atmosphere. If the system temperature and/or pressure are modified, or under the control of multi-configuration operands, be sure to adjust the wavelengths to the new air temperature/pressure. For more information, see "Wavelength data" under "Conventions and Definitions".

Wavelength Data Editor

The Wavelength Data editor is accessed by double clicking the "Wavelengths" bar in the System Explorer.
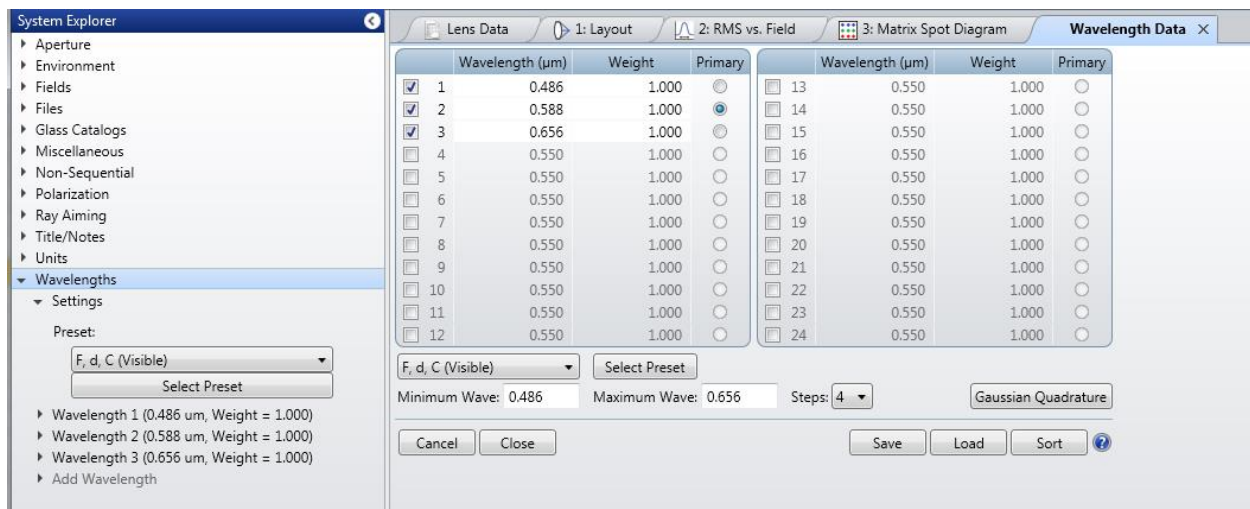
The Wavelength Data editor includes the same information and settings as the Wavelengths section of the System Explorer, but also includes a button to select wavelengths using a Gaussian Quadrature algorithm. This algorithm provides the most efficient choice of wavelengths and weights for optimization of an optical system subject to a broadband source. To use this approach, select the number of wavelengths (any even number from 2 to 12) and enter in the wavelength range (in microns), and then click on the "Gaussian Quadrature" button. The algorithm used here is described in B.J. Bauman and H. Xiao, "Gaussian quadrature for optical design with non-circular pupils and fields, and broad wavelengths", Proc. SPIE, Vol. 7652, p76522S-1 (2010). More details on the efficiency of this algorithm is provided in the Knowledge Base article entitled "Using Gaussian Quadrature to Model A Broad Spectrum".

The Save and Load buttons are used to store and retrieve the wavelength data independently from the lens data. The file format is text that can be edited or created outside of Zemax.

Environment

These Environment features are available in the System Explorer, and allow definition of the system temperature and pressure. Note that wavelengths are always measured in micrometers referenced to "air" at the system temperature and pressure. If the system temperature and air pressure change; care should be taken to adjust the wavelength definitions to match the new environment.

See Wavelengths and Thermal Analysis for more information. For more information on changing the temperature and pressure of individual surfaces see Defining multiple environments within a single configuration.
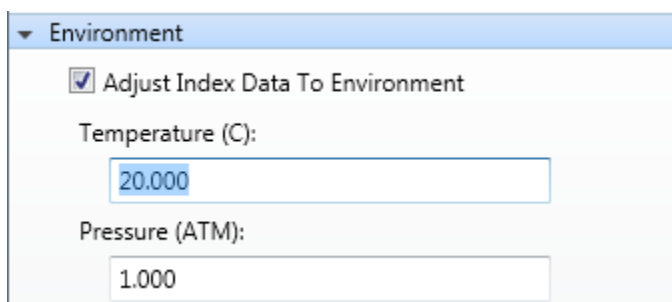
Settings:

Adjust Index Data To Environment If checked, all index data used for ray tracing will be adjusted from the "raw" glass catalog values to be relative to the system temperature and pressure. For details see Index of refraction computation. Individual surfaces may have a temperature and pressure that is different from the system temperature and pressure; for these surfaces the index data will be adjusted to reflect the environment of the surface.

If unchecked, all index data will be used directly from the glass catalogs, without any adjustment made for temperature or pressure changes or discrepancies between glass data defined for different temperatures. Warnings will be listed in the index data portion of the Prescription Data report (see Prescription Data) if this box is unchecked and the reference temperature of any glass differs from the system temperature by more than 6 degrees Celsius. Note that when the adjust index box is unchecked the system temperature is set to 20 degrees C and the pressure to 1.0 atmospheres, and therefore all index data must be relative to that environment.

It is highly recommended that "Adjust Index Data To Environment" always be checked "on". There is no disadvantage in leaving the switch on at all times, as long as all index data has been properly defined.

Temperature (C) The system temperature in degrees Celsius.

Pressure (ATM) The system air pressure in atmospheres. A value of 0.0 implies vacuum, 1.0 implies sea level.

Polarization

These Polarization features are available in the System Explorer. The controls in this section set the default input polarization state for many sequential analysis computations which use polarization ray tracing. For sequential analysis features such as Spot Diagrams and RMS vs. Field that support the "Use Polarization" switch to enable polarization ray tracing and apodization, this is the only method to set the initial polarization state.
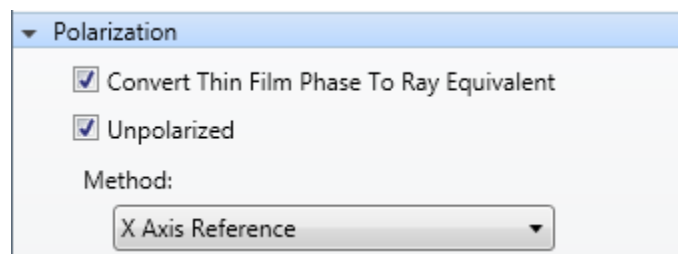
For most (but not all) sequential analysis features, the polarization ray trace is only used to determine the transmitted intensity of the ray while accounting for Fresnel, thin film, and bulk absorption effects. The rays are attenuated in intensity and a weighted computation is performed. Polarization phase aberrations and the vectorial nature of polarization are ignored.

Some features consider not only transmission, but also the separate orthogonal vector components of the polarized light, and polarization phase aberrations (these polarization aberrations can be calculated using the polarization phase aberration plot; for details, search the help files for "Phase Aberration"). The Huygens PSF and PSF Cross Section, Huygens MTF, and Encircled Energy using the Huygens PSF all consider the full polarization vector and polarization phase aberrations. These computations work by computing data for the Ex, Ey, and Ez components of the polarized electric field separately, then incoherently summing the results. The polarization phase aberrations induced in each orthogonal component of the electric field are considered as any other phase aberration. For a discussion of how Physical Optics Propagation considers polarization, search the help files for "Accounting for polarization".

To define the polarization state of non-sequential sources, see the "Sources" section of the Object Properties. For a detailed discussion of polarization and thin films, see Polarization Analysis.

Convert thin film phase to ray equivalent:

If selected, Zemax converts the polarization phase computed using thin film conventions to phase along the ray. If unselected, the ray coefficients will not be converted from the field coefficients. The recommended and default setting is to convert the field thin film phase to ray phase.

The thin film industry uses a different phase convention than is required for ray tracing. The thin film convention is to measure the phase shift along the normal vector as an imaginary plane wave propagates from the outermost coating layer to the substrate. This convention implies the phase shift is largest at normal incidence, and tapers off to zero by approximately the cosine of the angle of incidence for larger angles of incidence. Zemax calls the τ_s and τ_p values "Field" coefficients when using this reference.

For ray tracing, the optical phase advance or delay is measured along the ray. Zemax traces rays directly to the substrate, ignoring the coating thickness. This is because the thickness between surfaces specified in the Lens Data Editor is the thickness between the substrates, ignoring coatings. The coating is presumed to grow into the space preceding the surfaces. Correctly computing the phase of the ray requires back propagating the electric field to the point the coating begins, and adjusting the coating phase to be measured along the ray vector rather than the normal vector. Zemax calls

the τ_s and τ_p values "Ray" coefficients when using this convention. A similar computation yields the factors for reflective surfaces, or for surfaces where the coating stack has automatically been reversed, and Zemax automatically chooses the correct factors in all cases.

Unpolarized:

If checked, then the polarization values Jx, Jy, X-Phase, and Y-Phase are ignored, and an unpolarized computation will be performed. An unpolarized computation is performed by tracing two rays with orthogonal polarization and averaging the resulting transmission. Note that unpolarized computations take longer than polarized computations, which take longer than computations that ignore polarization entirely.

Some Zemax features, such as the optimization operand CODA, require a single specific polarization state to be defined. These features will use the defined Jx, Jy, X-Phase, and Y-Phase values defined below even if the "Unpolarized" option is checked.

Jx, Jy, X-Phase, Y-Phase:

These values specify the default input polarization state, and are only available if the user unchecks the "Unpolarized" box. The polarization is defined by four numbers: Jx, and Jy, which are the magnitudes of the electric field in X and Y, and the X-Phase and Y-Phase, which are the phase angles in degrees. Internally, Zemax normalizes the electric field vectors to have unity intensity. For an important discussion of the conversion from Jones vectors to 3D electric field, see Defining the initial polarization.

Method:

This feature is available whether or not the "Unpolarized" option is checked. It selects the method used to determine the S and P vectors based on the ray vector. For a discussion of this feature, see "Defining the initial polarization". The polarization is defined using a Jones vector:

$$J = \begin{bmatrix} J_x \\ J_y \end{bmatrix}$$

where Jx and Jy have both a magnitude and a phase, and the symbol J is used instead of E to distinguish the 2D Jones vector from the 3D electric field vector E. Zemax normalizes the specified *Jx* and *Jy* values to have unity magnitude, and then scales the intensity as appropriate if any pupil apodization has been specified. The Jx and Jy values are therefore measured in terms of relative electric field amplitude.

Let the ray vector be K, which has X, Y, and Z direction cosines (l, m, n). For rays traveling parallel to the Z axis, or K = (0, 0, 1), the electric field in the Z direction is zero, and the conversion from the Jones vector to the Electric field is trivial: Ex = Jx, Ey = Jy, and Ez = 0.

For a more general ray, the conversion of the Jones vector (Jx, Jy) to the 3D electric field values (Ex, Ey, Ez) is ambiguous. It is not possible to interpret the Jx and Jy values as meaning that for any ray, the Jx value should be applied in such a way to leave the Ey zero, and for the Jy to be applied so that the Ex is zero. The reason is that the E resulting for the Jones vectors (Jx = 1, Jy = 0) and (Jx = 0, Jy = 1) must be orthogonal to both K and to each other.

Zemax allows user selection of three different methods to perform the conversion from J to E. In each method, the vector K refers to the ray vector, the Jx value is the field along a vector S, and the Jy value is the field along a vector P. Note that K, S, and P must all be unit vectors and orthogonal to each other. The three methods are:

X Axis Reference The P vector is determined from K cross X, and S = P cross K. This method is the default.

Y Axis Reference The S vector is determined from Y cross K, and P = K cross S.

Z Axis Reference The S vector is determined from K cross Z, and P = K cross S.

When the object is at infinity, the method selected will change the polarization orientation of S and P for different fields, but all rays from the same field will have the same polarization since all rays are parallel to each other. For finite conjugates, especially when the object space numerical aperture is large, the S and P vector orientations will vary for different rays in the pupil. No matter which method is selected, the transmission results for unpolarized light will be unaffected because any two orthogonal rays may be traced to compute transmission. For systems where a particular polarization is desired, care should be taken to verify that the conversion from J to E yields the intended polarized ray. To review the detailed conversion results, see Polarization Ray Trace.

Advanced

These Advanced options are available in the System Explorer. Please read the documentation before changing these settings.
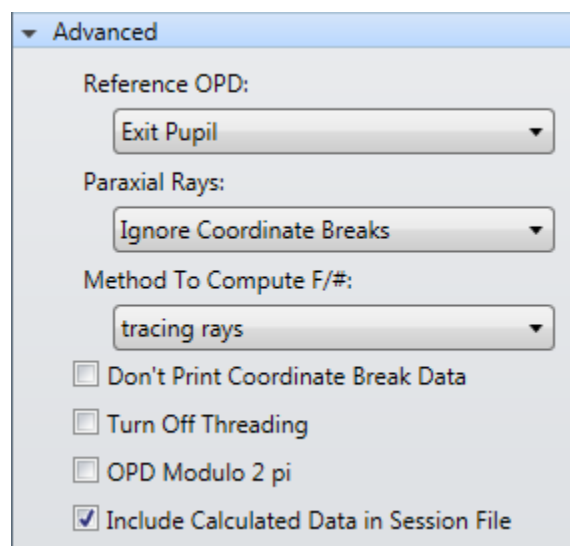
Reference OPD

The Optical Path Difference, or OPD, is of value in optical design computations because the OPD represents the phase error of the wavefront forming an image. Any deviations from zero OPD contribute to a degradation of the diffraction image formed by the optical system.

Because the exit pupil is the image of the stop in image space, the exit pupil represents the only location in image space where the beam has a clearly defined edge. The illumination at the exit pupil is generally smoothly varying in amplitude and phase, and there is a clearly defined boundary between regions of zero amplitude and nonzero amplitude. A reasonable assumption is that there are no diffraction effects apparent in the wavefront when viewed in the exit pupil. This is asymptotically true if all apertures in the optical system are large compared to the stop limited beam size incident upon each aperture. Even if the exit pupil is virtual, which is often the case, the exit pupil still defines the only location in image space where the beam is diffraction free.

As the wavefront propagates from the exit pupil towards the image surface, the beam profile becomes complex in amplitude and phase, and the wavefront extends over all space due to the effects of diffraction. For this reason, the phase error as measured in the exit pupil is uniquely and critically important to the description of the wavefront and image quality.

Exit Pupil Zemax by default uses the exit pupil as a reference for OPD computations. Therefore, when the OPD is computed for a given ray, the ray is traced through the optical system, all the way to the image surface, and then is traced backward to the "reference sphere" which lies in the exit pupil. The OPD as measured back on this surface is the physically significant phase error important to diffraction computations, such as MTF, PSF, and encircled energy. The

additional path length due to the tracing of the ray backwards to the exit pupil, subtracted from the radius of the reference sphere, yields a slight adjustment of the OPD called the "correction term". This computation is correct and is the desired method for all cases of practical interest.

Infinity The reference to "Infinity" makes the assumption that the exit pupil is very far away and that the OPD correction term is given strictly by the angular error in the ray. This setting should only be used if Zemax is unable to correctly compute the effective exit pupil location and size. This can occur with some very unusual optics which do not form images (real or virtual) of the stop surface, or for some non-axial systems where the computed exit pupil lies too close to the image surface for the exit pupil correction to be accurate. Do not use the Infinity reference unless there is compelling reason to believe that the exit pupil computation is not correct.

Absolute/Absolute 2 The reference to "Absolute" and "Absolute 2" means that Zemax does not add any correction term to the OPD computation. These methods are not usually physically significant or required for focal systems. For afocal systems, choosing "Absolute" will refer the OPD to a plane normal to the chief ray at the location of the image surface, regardless of the exit pupil location. This is a useful option if the exit pupil position cannot be determined accurately, as may be the case for non-axial afocal systems. The "Absolute 2" method is similar, except the OPD is not referenced to a plane normal to the chief ray.

Paraxial Rays

Ignore Coordinate Breaks Paraxial ray properties are generally not defined for non-rotationally symmetric systems. For this reason, Zemax by default ignores all tilts and decenters due to coordinate breaks when tracing paraxial rays. By ignoring tilts and decenters, Zemax can compute the paraxial properties of an equivalent centered system, which is generally the correct approach even for systems without symmetry. The default setting "Ignore Coordinate Breaks" is therefore highly recommended for systems where ignoring tilts and decenters results in a reasonable axial approximation to the actual system.

Consider Coordinate Breaks There are cases where "Consider Coordinate Breaks" may be required. For ray tracing through gratings, coordinate breaks may be required even for paraxial rays, otherwise, the rays may not be able to satisfy the grating equation. Ray tracing through non-sequential objects may also require that paraxial rays consider coordinate breaks.

Method to Compute F/#

Tracing Rays By default, Zemax computes the paraxial and working F/# of a system using ray tracing. For details, see the definitions for "Paraxial working F/#" and "Working F/#".

For systems with very large F/#s, the ray tracing method may be inaccurate, because the very small angles between the marginal and chief rays leads to round-off error, and even small amounts of aberrations such as spherical aberration can significantly affect the calculation of F/#. For these reasons, Zemax will "cap" the maximum F/# to a value of 10,000. It is recommended that rays be used to compute the F/# for all systems whose F/# is less than 10,000.

Pupil Size/Position The preferred method of modeling systems with very large F/#s is to use afocal mode. See "Afocal Image Space". Optionally, the F/# can be computed using the exit pupil distance divided by the exit pupil diameter. There are some important points to consider when using this option. First, Zemax makes no attempt to scale the F/# with either pupil orientation or field angle. Second, the exit pupil diameter and position used are based upon the paraxial values for these parameters (see the definition for "Exit pupil diameter"). In the presence of significant pupil aberration, the paraxial values may not agree with the real, aberrated pupil sizes. The best way to avoid this possibility is to place the stop at the end of the optical system, prior to the image surface, and use ray aiming (see the definition for "Ray Aiming") and float by stop size for the aperture type (see the Aperture section of the Surface Properties). This option is also useful when the on-axis field cannot be traced and the axial F/# is different from the actual F/# at the fields being traced.

Other Settings:

Don't Print Coordinate Break Data

If checked, selected data will not be printed for coordinate break surfaces. Use of this option can shorten and clarify some text listings, particularly in systems with many coordinate break surfaces. Supported features include ray trace listings, index data, and global coordinate data. Support for this option in other Zemax features may be added upon request.

Turn Off Threading
If checked, Zemax will not split calculations into multiple threads of execution. Threading allows computers with multiple CPU's or multi-core CPU's to calculate results more quickly. The only reason to turn off threading is if insufficient memory exists to break calculations into separate threads. This is primarily a debugging and diagnostic tool and should normally be left unchecked.

OPD Modulo 2 pi
If checked, all OPD data will be computed as the fractional part of the total OPD. All OPD computations will return results that are between -π and +π, or -0.5 and +0.5 waves. This method of computing OPD may have applications to some specific problems. However, its use is discouraged unless the user fully understands the meaning and ramifications of computing OPD in this manner.

Include Calculated Data in Session File
If checked, then calcualted data for all open analysis windows (sequential mode) and/or all detectors (non-sequential mode) will be cached in the current session file. This will allow the data to be restored when the lens file is loaded – significantly decreasing the time for file load – but also increases the size of the session file associated with the lens file. Note that this setting overrides the default "Include Calculated Data in Session File" setting in the Project Preferences.

Ray Aiming

These Ray Aiming features are available in the System Explorer. The controls define the ray aiming algorithm.
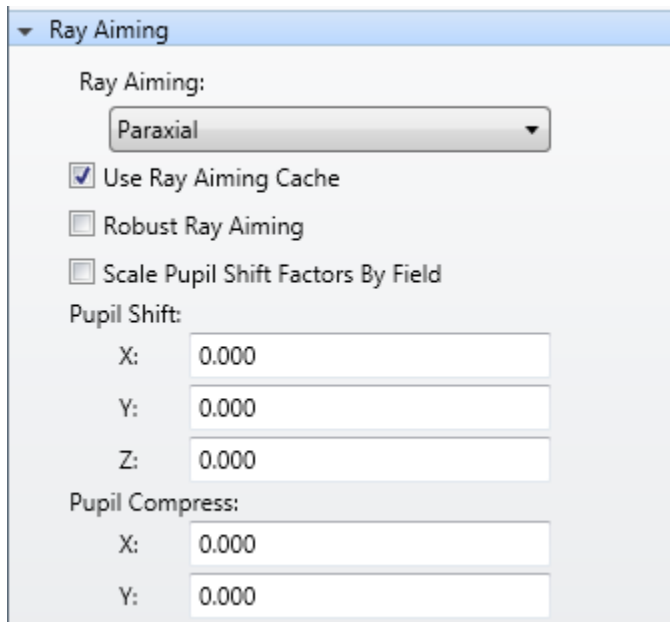
Ray Aiming:

Ray aiming is an iterative ray tracing algorithm in Zemax that finds rays at the object that correctly fill the stop surface for a given stop size. It is generally only required when the pupil, which is the image of the stop as seen from object space, is considerably aberrated, shifted, or tilted.

Ray Aiming Off If ray aiming is off, Zemax will use the paraxial entrance pupil size and location determined by the aperture settings and calculated at the primary wavelength on axis for launching rays from the object surface. This means Zemax will ignore entrance pupil aberration. For slow systems with modest field angles, this is perfectly acceptable. However, certain systems, such as those with low F/#'s or large field angles, may have significant entrance pupil aberration. The two primary effects of pupil aberration are the shift in location of the pupil with field angle, and the anamorphic scaling of the edges of the pupil.

Zemax can be instructed to account for aberrations of the pupil using ray aiming. With ray aiming, every ray trace is performed iteratively, with the program adjusting the ray coordinates or cosines in object space so that the ray crosses the correct location on the stop surface. To determine the correct location on the stop surface, the stop surface radius must be computed. The stop surface radius is computed by tracing a marginal ray from the center of the object to the stop surface at the primary wavelength. Either paraxial or real rays may be used in this trace to determine the stop radius.

Paraxial Ray Aiming Paraxial rays are well behaved and paraxial definitions are commonly used for most first-order system properties such as focal length, F/#, and magnification, and so paraxial rays may also be used to determine the stop size. However, for systems that have significantly aberrated pupils, there will be a difference between the paraxial and real ray stop radius. These systems will exhibit a difference between the real and paraxial ray system aperture. For example, the paraxial object space numerical aperture may be defined as 0.4, but the actual numerical aperture of real rays may be a different value.

Real Ray Aiming For real rays to have the object space properties defined by the system aperture, use real rays instead of paraxial rays to determine the stop radius. Note that real ray based ray aiming will not work in systems where the stop lies in a caustic or where the real rays cannot be traced at the full entrance pupil diameter or numerical aperture. If real ray aiming causes any of these problems, set the ray aiming to paraxial rather than real. Note that once the stop radius is determined all rays are aimed to the correct location on the stop, regardless of whether paraxial or real rays were used to determine the stop radius.

Although real ray aiming is more accurate than paraxial entrance pupil aiming, most ray traces will take from two to eight times as long to perform. Therefore, real ray aiming should only be used when required, and when paraxial ray aiming does not account for the majority of pupil aberrations.

To determine the amount of entrance pupil aberration in your system, select ray aiming off, and then look at the pupil aberration plot. Pupil aberration of less than a few percent is generally insignificant. If your system has significant pupil aberration, turn paraxial ray aiming on, and look at the pupil aberration plot again. If there is still significant pupil aberration with paraxial ray aiming, then change to the real ray aiming algorithm. The aberration will decrease to zero, or very nearly so. Note that ray aiming does not, of course, actually eliminate pupil aberration, it merely accounts for it.

To eliminate any ambiguity in the calculation of the actual stop size, set ray aiming to paraxial or real, and then set the system aperture type to "float by stop size". This eliminates the need for any ray tracing at all to determine the stop size, and both real and paraxial rays will be aimed to the real stop exactly.

For systems with virtual stops, such as some eyepieces, the effective stop location and size may be a function of wavelength. For these systems, use the multi-configuration capability to treat each wavelength and system aperture definition separately.

Use Ray Aiming Cache:

If checked, Zemax caches ray aiming coordinates so that new ray traces take advantage of previous iterations of the ray tracing algorithm. Using the cache can speed up ray tracing dramatically. However, use of the cache does require that the chief ray can be traced accurately. For some systems, the chief ray cannot be traced, and for these systems, the cache should be turned off.

Robust Ray Aiming (slow):

If checked, Zemax uses a more reliable, but slower algorithm for aiming rays. This switch should only be set if the ray aiming algorithm is failing even with the cache turned on. This switch has no effect unless the ray aiming cache is checked on. Robust mode goes through an additional check to make sure that if multiple ray paths to the same stop surface location exist, only the correct one is chosen. This is typically a problem in very fast, very wide angle systems where off axis fields may find a virtual path to the stop that confuses the ray aiming iteration.

Pupil Shift:

For some very wide angle or highly tilted or decentered systems, the ray aiming feature will fail if unassisted. The problem is that the paraxial entrance pupil is used as a first guess to trace the ray. If the pupil aberration is severe, it is possible that even this first guess cannot be traced, which prevents the algorithm from taking a second, more refined guess.

The solution is to provide a rough guess as to how much the pupil has been shifted and compressed with respect to the paraxial pupil. There are three shift components; x, y, and z; all measured in lens units. There are two compress components; x and y, and these are dimensionless scaling factors. The default value of zero for all five may be modified to assist the algorithm in finding a successful first guess for the ray aiming.

The shifts move the center of the aim point on the paraxial entrance pupil. Positive values for the z shift indicate that the aim point is to the right of the paraxial pupil, negative values indicate the pupil is shifted to the left. Most wide angle systems have left-shifted pupils. The z pupil shift value provided is scaled linearly with the field angle of the ray being traced, so the pupil shift refers to the offset of the pupil at full field. All shifts are in lens units.

Scale pupil shift factors by field:

If selected, the x and y pupil shift values are also scaled with field, otherwise, the x and y shift values are used for all fields without any scaling. All shifts are in lens units.

Pupil Compress:

The x and y compress values are used to change the relative coordinates on the paraxial entrance pupil to start the iteration. A value of zero means no compress, while a value of 0.1 indicates the pupil is compressed 10%. The compress values are particularly useful when the real pupil is smaller than the paraxial pupil, and rays traced at the full paraxial pupil size are difficult or impossible to trace.

It is important to understand that the exact values of the pupil shift and compress values is unimportant. Once the first guess ray can be traced, the algorithm will robustly find the exact pupil location. The pupil shift and compress values are just to get the ray aiming started. Neither the shift nor the compress values actually change the size of the entrance pupil. Generally, guessing at the pupil shift and compress values is an acceptable way of determining a suitable value.

Material Catalogs

These Material (or Glass) Catalog options are available in the System Explorer. To access a glass catalog, it must be placed in the glass catalog folder (see Folders). For more information on changing glass catalogs, see Using Glass Catalogs. To set a default catalog, see the Editors section of the Project Preferences window, accessible from the Setup Tab.
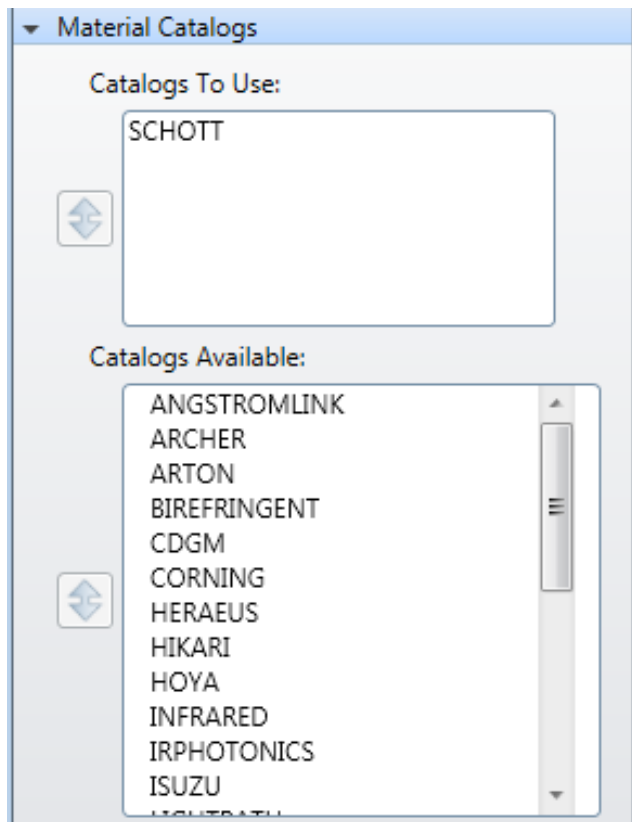
Settings:

Catalogs To Use Lists the names of the currently used glass catalogs (without file extensions).

Catalogs Available Lists the names of the available but not currently used glass catalogs. Catalogs may be moved back and forth between these lists using the arrow buttons.

Non-Sequential

These Non-Sequential features are available in the System Explorer. The controls in this section define how rays trace within a NSC group.

Maximum Intersections Per Ray:

This control defines how many times a single ray may intersect objects along any single path from the original source parent ray to the final object intersection. When ray splitting is used, this parameter controls the maximum number of generations of child rays split off from the parent ray.

In some systems, such as a source placed inside a reflective sphere with perfect reflectivity and no bulk absorption, the rays will bounce around until this limit is reached, and the energy of the ray finally will be discarded. These systems are of course not physically possible, since no reflection or propagation through anything but a vacuum is perfectly lossless. The maximum value allowed is 4,000 intersections.

Maximum Segments Per Ray:

This is the maximum number of segments *per ray launched*, not the total number of segments that Zemax can keep track of. A segment is that portion of a ray path from one intersection to the next. When a ray is launched from the source, it travels to the first object. That is 1 segment. If the ray then splits into 2 rays, each of those are another segment (for a total of 3). If each of those rays split again, there will be 7 segments. Generally, if ray splitting is being used, the number of segments grows far faster, and needs to be set much larger, than the number of ray-object intersections does.

It is critical to understand that *each* ray launched from a source can have up to the maximum number of segments. If the maximum number of segments is 1,000, and 5,000 individual rays are launched, Zemax can store up to 5,000,000 segments. The total RAM requirement is about 140 bytes times the maximum number of segments. Setting the limit at 100,000 segments would require 14 Mb of RAM *per ray*. For this reason, do not arbitrarily (and needlessly) set the maximum to very large numbers; there is a limit of 2,000,000 Zemax will allow.

Note that with even a modest number of segments, Zemax can still trace millions of rays.... the maximum number of segments is only a limit on how many segments are allowed in *one single ray*. Most Zemax features only use one ray at a time, so the total RAM requirement is driven by the maximum number of segments rather than the number of rays traced. The exception is the Shaded Model display, which must be able to allocate enough RAM to hold all segments for all source rays drawn.

Maximum Nested/Touching Objects:

This defines an upper limit on how many objects can be inside, straddled, or in direct contact with each other. For example, if object 3 is inside of 2 which is inside of object 1, the maximum number of nested objects is 3. There may be any number of groups of objects each nested 3 deep in this case. The limit applies to the total nesting in any one collection of objects, however, there may be any number of such collections within the NSC group. If several objects share a common boundary, such as multiple volumes touching at one face, then the maximum nested objects must be set at least as high as the total number of objects that share a common point in space.

Maximum Source File Rays In Memory:

This parameter sets the maximum number of rays for each Source File object that will be held in memory. The recommended value is 1,000,000 rays; the minimum value is 5000 rays. For a complete description of how this setting affects memory usage and ray randomization see "Source File" under the reference file for Non-Sequential Sources.

Minimum Relative Ray Intensity:

As each ray splits, the energy decreases. The relative ray intensity is a lower limit on how much energy the ray can carry and still be traced. This parameter is a fraction, such as 0.001, relative to the starting ray intensity from the source. Once a child ray falls below this relative energy, the ray is terminated.

Minimum Absolute Ray Intensity:

This parameter is very similar to the minimum relative ray intensity, except it is absolute in source units rather than relative to the starting intensity. If this is zero, the absolute ray intensity threshold is ignored. The initial intensity of each ray is always given by the source intensity divided by the total number of analysis rays for that source. The number of layout rays is not used to determine the initial intensity of rays, even for those rays drawn on layout plots.

Glue Distance In Lens Units:

When two NSC objects are placed in contact, such as a lens touching one face of a prism, numerical roundoff will cause the ray tracing algorithm to sometimes detect a very tiny distance between the two objects. This can also occur when objects are rotated in 3D space and placed close, but not exactly, next to one another because of the finite number of digits entered in the spreadsheet editor.

The glue distance is the distance below which the objects are considered in contact. It is important that objects not be separated by distances very close to the glue distance. If it is intended that two objects be in contact, then the maximum spacing between the objects should be several times smaller than the glue distance. If it is intended that the objects be separated, then the distance between the object should be several times larger than the glue distance. Object spacings very close to the glue distance will yield inconsistent ray tracing or geometry errors. This should be avoided by adjusting either the object spacing or the glue distance.

The glue distance also determines the minimum propagation length for ray tracing. If a ray-object intersection is less than the glue distance away from the previous intercept, the intercept is ignored.

The glue distance is also used in the tolerance associated with ray tracing to general curved surfaces. Zemax will iterate until the error in the ray-surface intercept solution is less than one-fifth of the glue distance.

In the majority of cases, no adjustment should be made to the glue distance parameter. The glue distance must be no smaller than 1.0E-10 and no larger than 1.0E-03.

Missed Ray Draw Distance In Lens Unit:

This parameter is the distance to draw the ray segments that miss all objects. Zemax will draw a short ray segment to indicate the direction the ray was traveling in. This parameter also controls how large a source indicator arrow to draw. If zero, Zemax will select a default value for this parameter when drawing miss rays and some sources.

Simple Ray Splitting:

When a ray strikes a refractive boundary, generally some energy will reflect and some will refract. If Simple Ray Splitting is off, then both the reflected and refracted rays are traced. Each ray gets the fraction of energy corresponding to the reflection and transmission coefficients of the interface. If Simple Ray Splitting is on, then either the reflected or the refracted ray is traced, but not both. The decision to trace the reflected or the refracted ray is random; with the reflection and transmission coefficients being interpreted as a relative probability of taking that path. Whichever path is selected, the reflected or refracted ray gets all of the energy that would have propagated down both paths. The advantage to using Simple Ray Splitting is fewer rays traced, so computations are faster. The disadvantage is the rays traced contain less detailed information. For more information search the help files for "Ray splitting".

Retrace Source Rays Upon File Open:

If checked, then NSC source rays will be retraced when the file is opened. This allows refreshing of detector windows automatically.

Title/Notes

The Title/Notes fields are available in the System Explorer.
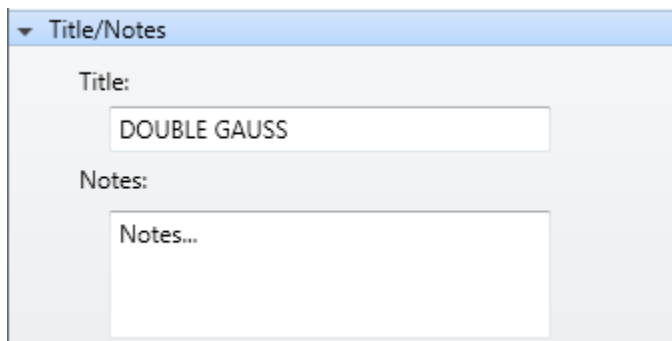
Settings:

Lens Title The lens title will appear on graphic and text output. The title is specified by typing the title in the required space.

Notes The notes section allows entry of a few lines of text which is stored with the lens file.

Files

These file options are available in the System Explorer. This feature is used to select the file in the Zemax directory that contains the relevant data for this lens.
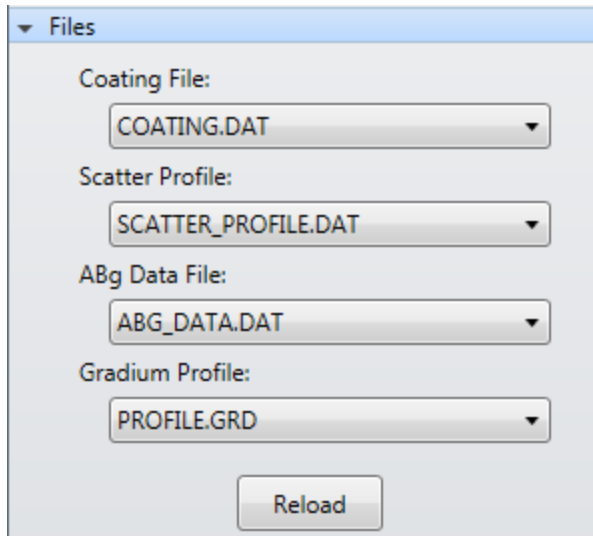
Settings:

Coating File The name of the file, in the coatings folder (see [Folders](#)), that contains the coating material and layer definitions used by this lens. The default name is COATING.DAT. Each lens file may use a separate coating file if desired.

Scatter Profile The name of the file, in the <data>\Profiles folder (see [Folders](#)), that contains the scattering profiles used by this lens. New scatter profiles may be added or old ones deleted on the Scattering tab of the NSC Objects property dialog box. See "Coat/Scatter tab" for details. The default name is SCATTER_-PROFILE.DAT. Each lens file may use a separate scatter profile file if desired.

ABg Data File The name of the file, in the <data>\ABg_Data folder (see [Folders](#)), that contains the ABg data definitions used by this lens. The default name is ABG_DATA.DAT. Each lens file may use a separate ABg data file if desired.

Gradium Profile The name of the file, in the glass catalog folder (see [Folders](#)), that contains the Gradium surface profile data. The default name is PROFILE.GRD. Gradium profile files must end in the extension GRD. See "GRADIUM™".

Units

The Units settings are available in the System Explorer.

Lens Units:

Lens units defines the units of measure for dimensions in most of the spreadsheet editors. These dimensions apply to data such as radii, thickness, entrance pupil diameters, non-sequential position coordinates, and most other parameters in Zemax.

There are four choices for lens units: millimeters, centimeters, inches, or meters.

For most imaging analysis features, such as ray fans and spot diagrams, the displayed units are micrometers. The symbol for micrometers is µm. The term "microns" is also commonly used in optics. The unit micron is identical to micrometer. Wavelengths are always defined in micrometers. The choice of lens units has no effect on the wavelength units.

Source Units:

Source units define the unit of measure for the flux (power) or energy emitted by non-sequential sources. This setting is used for sources defined in the non-sequential component editor, and for defining the power and irradiance in physical optics analysis. Source units may be either watts, lumens, or joules, with the additional choice of the prefixes femto, pico, nano, micro, milli, kilo, mega, giga, or tera. Watts are used for radiometric analysis, lumens for photometric analysis, and joules for energy analysis. The key difference between radiometric and photometric units is that photometric units are wavelength weighted to the response of the human eye. See the table in the next section for more information on units.

Analysis Units:

Analysis units define the unit of measure for irradiance (radiometric) or illuminance (photometric). This setting only affects data as displayed on detectors collecting light from sources defined in the non-sequential components editor. Irradiance units are watts/area, where area is in square meters, centimeters, millimeters, feet, or inches. Illuminance units are lumens/area. Energy density units are joules/area. The unit prefixes femto, pico, nano, micro, milli, kilo, mega, giga, or tera are all supported.

One unfortunate convention in the optics industry (and in Zemax) is to use the term "Intensity" for two distinctly different things. Intensity is commonly used in ray tracing applications and the optical design community to define the flux represented by a single ray, measured in watts or lumens. Intensity is also used in the illumination and radiometry fields, but the definition used there is flux per solid angle, measured in watts per steradian or lumens per steradian (candela).

Zemax keeps track of energy by associating with each ray a flux value, which is equal to the square of the electric field vector also associated with the ray. When this ray strikes a detector object, Zemax computes the area the pixel represents to compute irradiance (flux/area) and the solid angle the pixel represents to compute the radiometric intensity (flux/solid angle). The difference in these two uses of the word intensity can always be resolved by considering the nature of the analysis Zemax is performing.

For more information on radiometric and photometric units, see the Handbook of Optics referenced in "What doesn't Zemax do?"

Afocal Mode Units:

Afocal mode units may be microradians, milliradians, radians, arc-seconds (1/3600 of a degree), arc-minutes (1/60 of a degree), or degrees. For more information on afocal mode, search the help files for "Afocal Image Space".

MTF Units:

MTF units for focal systems may be cycles per millimeter or cycles per milliradian. When using cycles per millimeter, the MTF is computed for spatial frequencies in image space on the image surface. When using cycles per milliradian, the MTF is computed for angular frequencies in object space. The MTF units selection affects the units for all MTF computations, including analysis, optimization, and tolerancing. The MTF units selection is not used in afocal mode, as afocal mode MTF always uses cycles per afocal mode unit.

Project Preferences

The Project Preferences are available in the System section of the Setup tab. The Project Preferences button opens a window where the user can define preferences for the current project and save to a project configuration file.

Address

The Address settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab. These settings determine how the "address" box will appear. The address box can be used to

display user defined text such as a company name or drawing number. The address box appears on the bottom right corner of most graphics.
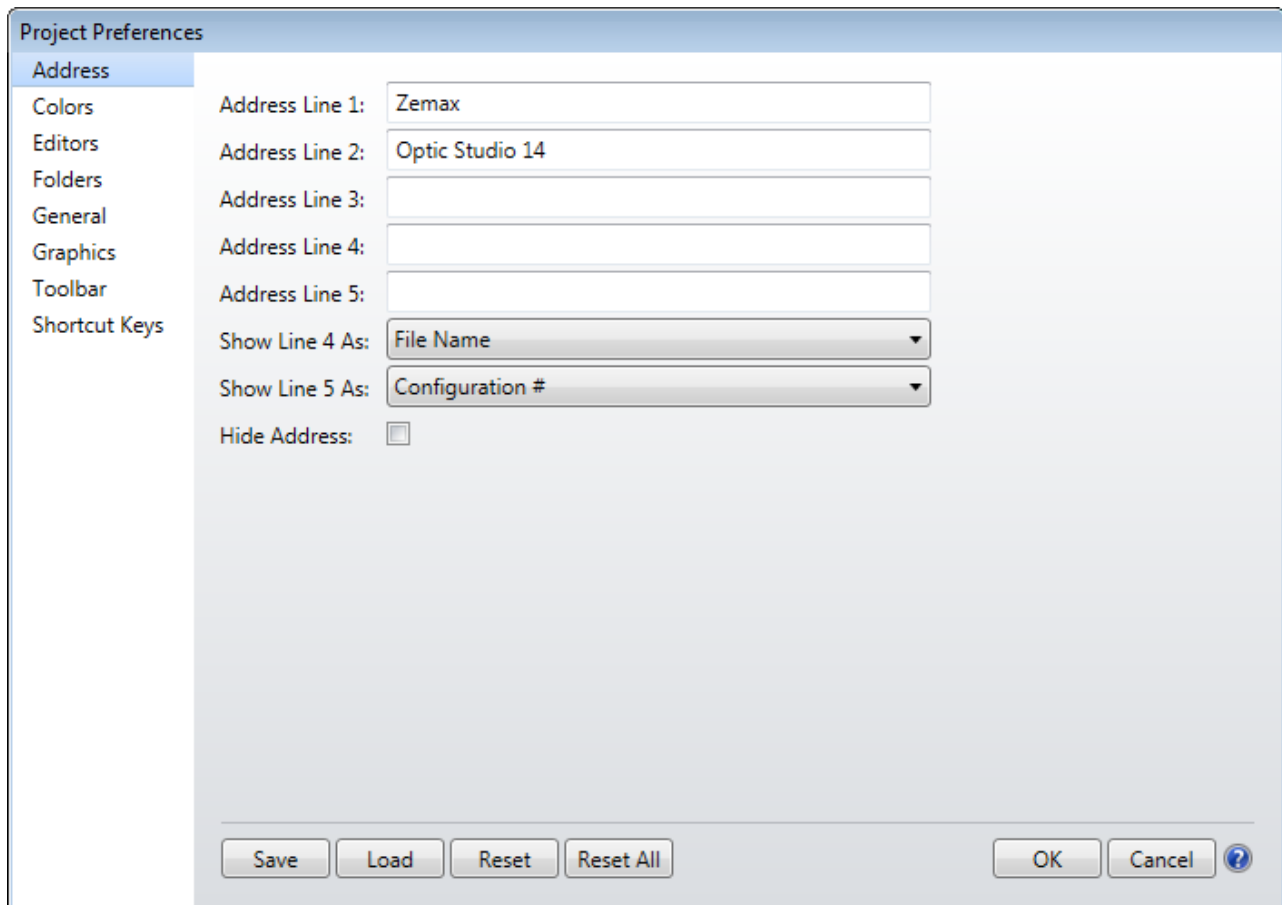
Settings:

Address Line 1 The first line of text to appear in the address box.

Address Line 2 The second line of text to appear in the address box.

Address Line 3 The third line of text to appear in the address box.

Address Line 4 The fourth line of text to appear in the address box, unless file name or zoom position is chosen instead.

Address Line 5 The fifth line of text to appear in the address box, unless file name or zoom position is chosen instead.

Show Line 4 As Choose either the entered text, the lens file name, or the zoom position.

Show Line 5 As Choose either the entered text, the lens file name, or the zoom position.

Hide Address If checked the address box will not appear.

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Colors

The Colors settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab. These settings define the colors used for drawing graphics and for the background color of rows in the various spreadsheet editors.
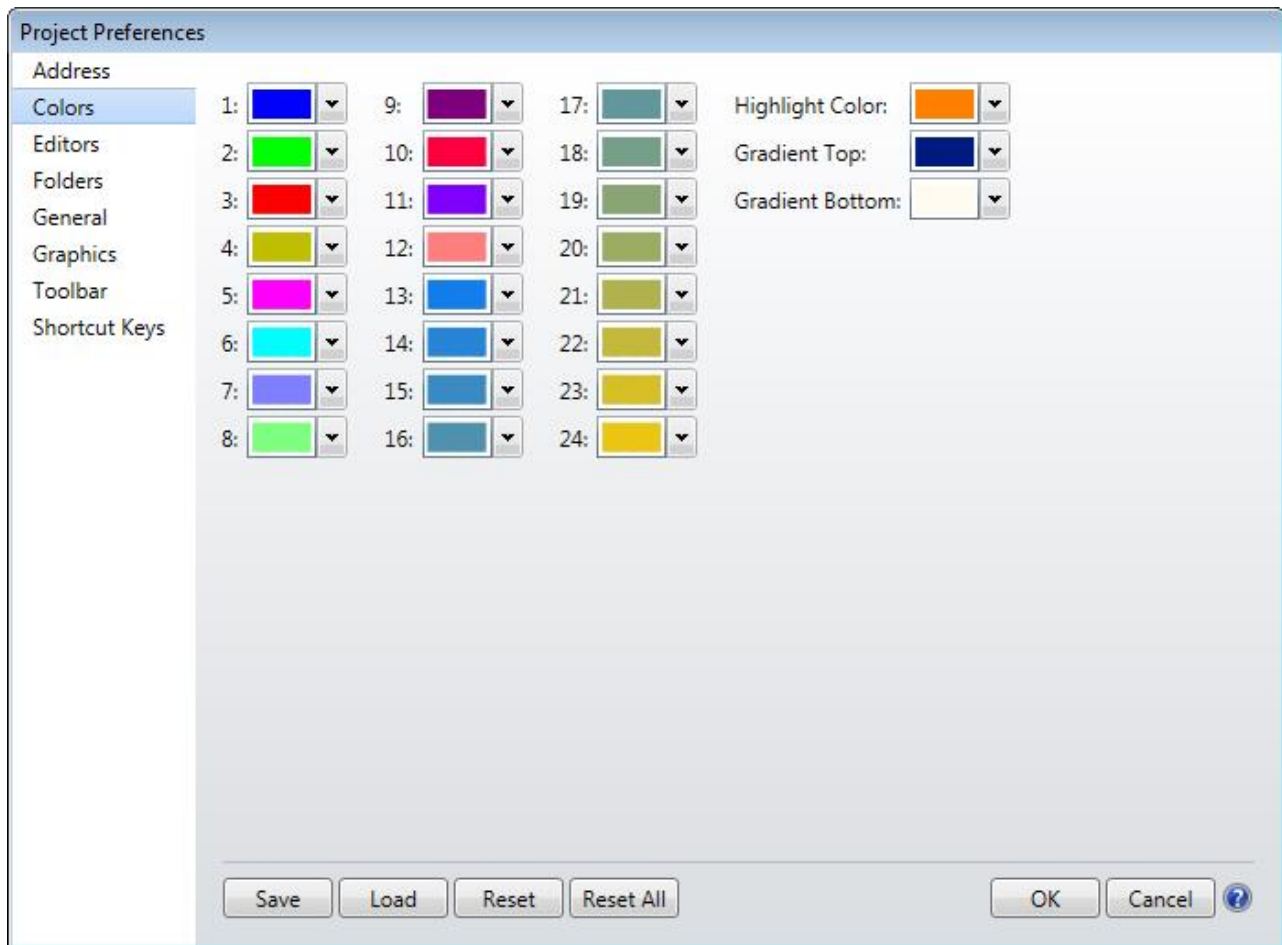
Settings:

Each color is used by the corresponding wavelength or field position when graphing ray fan data, spot diagrams, and other data. For example, typically wavelength or field number 1 uses color 1, field or wavelength 2 uses color 2, and so on. When using the defined colors in the spreadsheet editors, Zemax automatically blends the color with white to make the foreground text more readable.

Highlight Color The color Zemax uses to draw lens surfaces and NSC objects currently selected in the LDE and NSCE, respectively.

Gradient Top/**Gradient Bottom** These colors are used to define the colors for the gradient fill background in the Shaded Model displays.

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Editors

The Editors settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab. These settings determine characteristics of the spreadsheet editors. If the editor cell sizes are too narrow to display all the data in each cell, an asterisk "*" symbol will be printed instead of the truncated data.
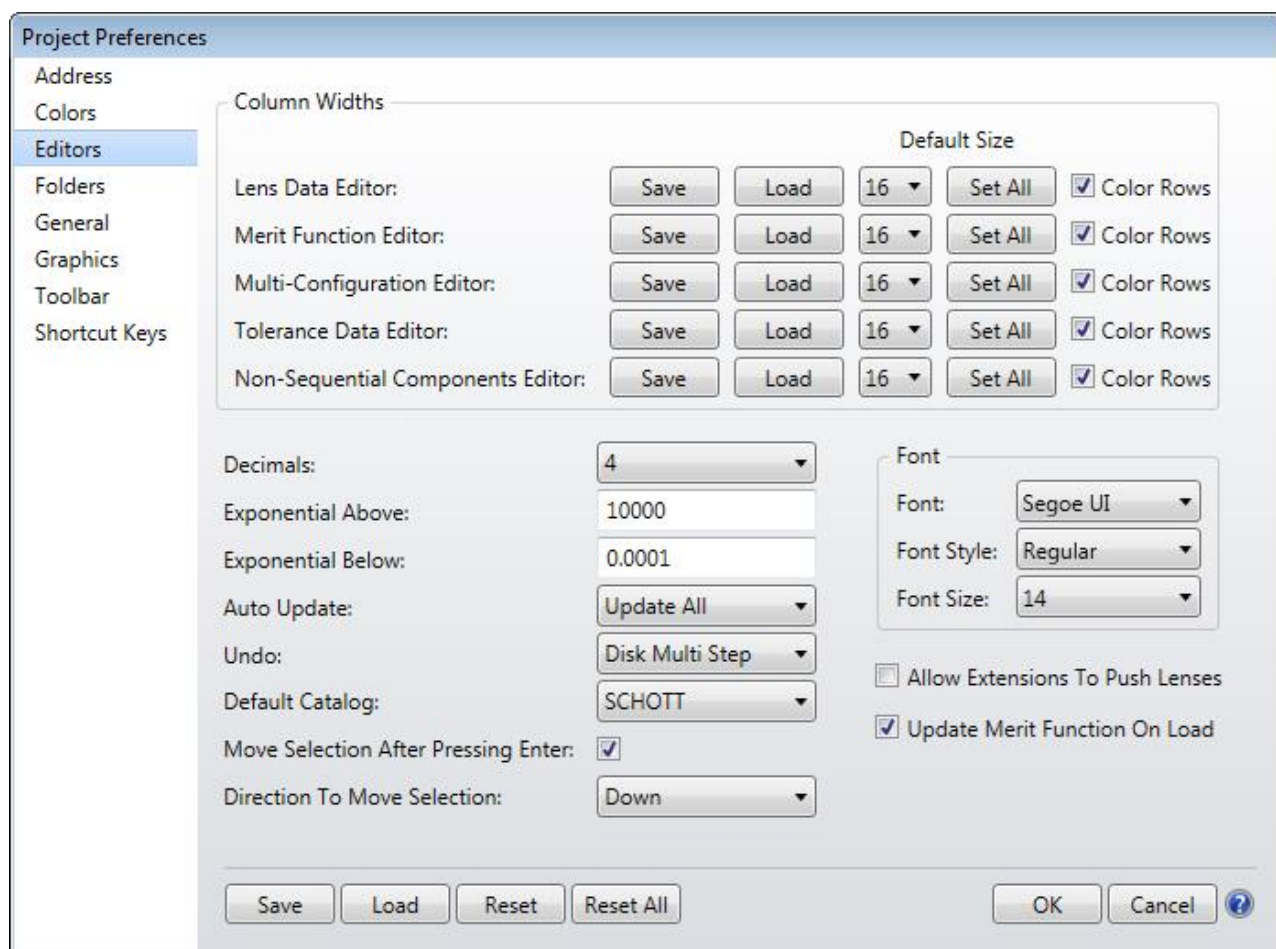
Individual Editor Settings:

Save Saves the current width settings of the specified editor.

Load Reloads the last saved width settings of the specified editor.

Cell Size The width in characters of a single edit cell in the specified editor.

Set All Applies the user selected cell size to all cells in the editor.

Color Rows If checked, then rows in the specified editor spreadsheet are color coded. See "Row Color".

Other Settings

Decimals This sets the number of decimal digits displayed in the editors. Selecting "Compact" will vary the number of decimals displayed to minimize the space required to display numbers.

Exponential Above / Below Zemax displays numeric data in either fixed decimal or exponential (scientific) notation. If the absolute value of the number is either a) greater than or equal to the "Exponential Above" value, or b) not zero but less than or equal to the "Exponential Below" value, then the number is displayed in exponential notation. Otherwise fixed notation is used.

Auto Update Controls how and when Zemax performs updates on the editor data. The "None" setting means pupil positions, solves, and other lens data editor data are not updated until "Update" button is pressed selected from the desired data window. The "Update" setting will cause an update to be performed whenever new data is typed in to the lens data or multi-configuration editors. "Update All" causes all windows to be updated whenever new data is typed in the editors.

Undo The Undo feature has three selectable states: None, Memory 1 Step, and Disk Multi Step. For more information, see "Using the Editors".

Default Catalog The name of the default glass catalog to use. See "Glass Catalogs".

Move Selection After Pressing Enter / Direction To Move Selection After pressing Enter in an editor cell, this setting specifies which adjacent cell is selected next. The options are: down, right, up, and left.

Font Choose the Font, Style, and Size for the editor displays. To change the text in the graphics based analysis windows see the "Graphics" section of the Project Preferences. To change the text displayed in the text based analysis windows see the "General" section of the Project Preferences.

Allow Extensions To Push Lenses If checked, then any Zemax extension can replace the lens data in the Lens Data Editor with new data, without warning or prompt. This option should not be checked unless it is intended that an extension control Zemax. See "Extensions" for more information on extensions.

Update Merit Function On Load Controls when Zemax performs updates on the merit function editor data.

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Folders

The Folders settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab. These settings determine where Zemax will place or search for certain types of files.
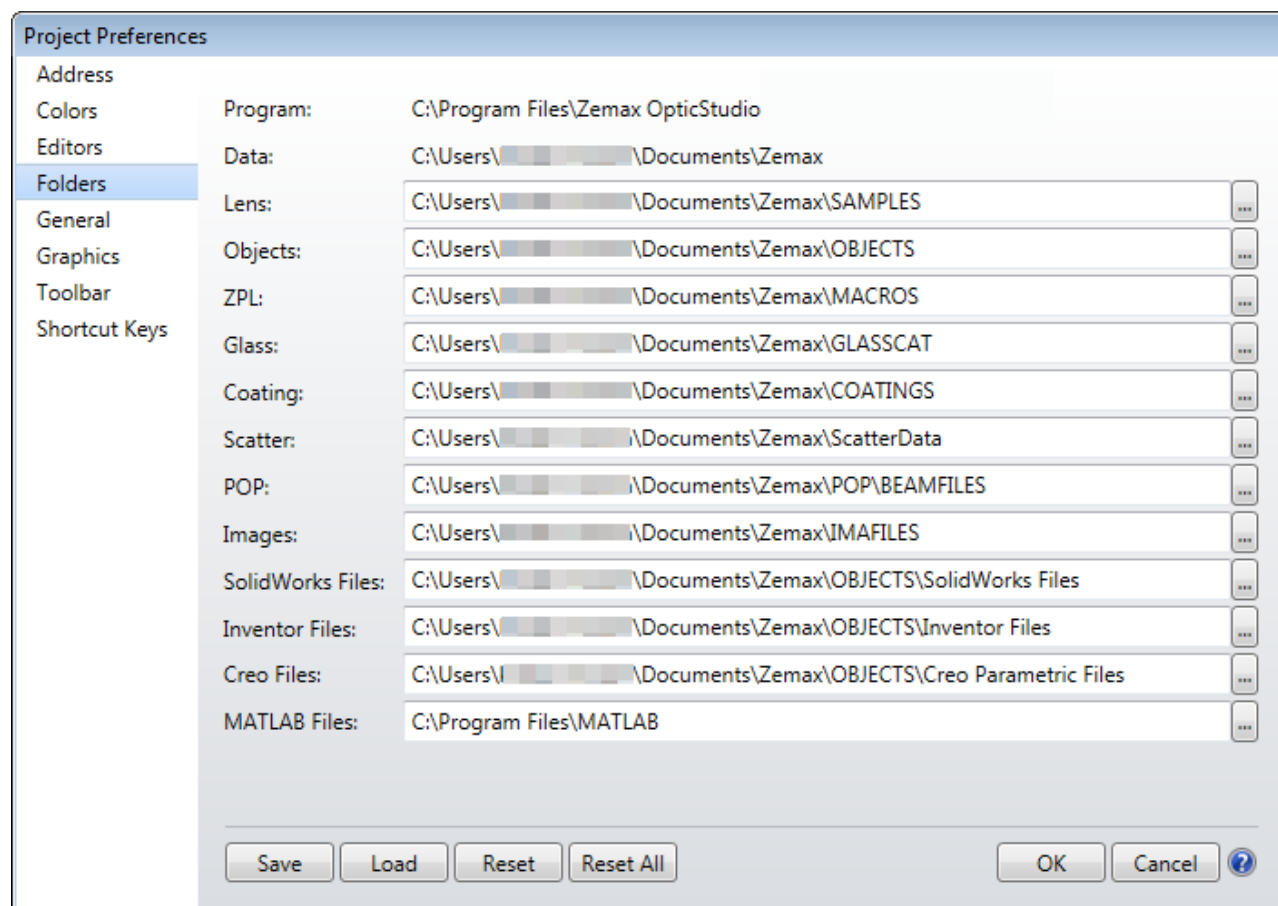
Zemax uses two main folders for storing programs and data. The Zemax program, Dynamic Link Library (DLL) files both supplied with Zemax and added for user-defined capabilities, and executable files for Zemax extensions, are stored under the Program Files folder, typically in C:\Program Files\Zemax. This folder is determined upon installation and should generally not need to be changed by the user. This folder is called the "Program" folder and is referred to as <program>.

All other user data is by default stored in the My Documents\Zemax folder, where "My Documents" is generally operating system dependent. This folder will usually be C:\Documents and Settings\Username\Documents\Zemax on Windows XP and earlier versions of Windows, or C:\Users\Username\Documents\Zemax on Vista and Windows 7. Zemax will display this root folder as the "Data" folder, and is referred to as <data>. This folder name and location cannot be modified.

The folders for lens, object, macro, glass catalog, coating, POP, image, scatter, CAD, and MATLAB files may optionally be modified from the default using these preference settings. The "?" button may be used to browse for and to select the

desired folder. Note that if the default folder names are modified and alternate folders used, then any updated files distributed with Zemax will NOT automatically be updated in the new folder names. For this reason, it is recommended that if the folder names are modified, that none of the files included with the Zemax distribution (such as SCHOTT.AGF) are used in the modified folders.

Settings:

Lens The folder for lens files.

Objects The folder for Non-Sequential Component object, source, aperture, and other files.

ZPL The folder for ZPL Macros. Macros may also be placed in subfolders within the specified folder.

Glass The folder for glass catalog files.

Coating The folder for coating definition files.

Scatter The folder for scatter data files.

POP The folder for POP ZBF beam files.

Images The folder for Image Simulation IMA, BMP, JPG, and PNG files.

SolidWorks Files The folder for SolidWorks *.SLDPRT and *SLDASM files.

Inventor Files The folder for Autodesk Inventor *.IPT and *.IAM files.

Creo Parametric Files The folder for Creo Parametric *.PRT and *.ASM files.

MATLAB Files The folder where the MATLAB executable is installed.

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Zemax file types by extension

The following table lists some of the extensions used by Zemax to indicate different types of data files.

FILE TYPES BY EXTENSION

Extension	Description

AGF	ANSI Glass Format. These are text files which contain the data for the glass catalogs. A text format is used so that additional data may be incorporated in future releases of Zemax.
ANN	Annotation files. These binary files store user defined annotations for Zemax graphics.
BGF	Binary Glass Format. To accelerate the loading of AGF files, Zemax converts AGF files to BGF, which are the version specific binary images of the glass catalogs.
C	C language source code files.
CFG	Configuration file. Zemax.CFG is the main configuration file which describes the user-selected options on the environment dialog box. There are numerous other files that may be present which end in CFG, such as RAY.CFG, which contains the user-defined defaults for the ray fan plot.
DLL	Dynamic Link Library files. These files are externally compiled programs linked into Zemax at run time. Typically, DLLs are used to implement user-defined surfaces, objects, or other features.
EXE	Executable file. This includes the main program, Zemax.EXE and other executables and utilities used by Zemax.
IGS, IGES	Initial Graphics Exchange Specification (IGES) file.
IMA	Image file, used by the image analysis feature.
NOT	Element drawing note files.
POB	Polygon object definition file for NSC object.
SES	These files are the Zemax "session" files, which define the display configuration and open windows and settings for each saved ZMX lens file.
STP, STEP	STEP files are CAD format files for describing solid model data.
TPD	Test Plate Data files. These contain test plate lists supplied by various optical fabricators, and are used in the test plate fit feature.
ZBF	Zemax Beam Files. These are used by the Physical Optics Propagation feature.
ZAR	Zemax archive files. Includes the *.ZMX lens file, and all associated files in single archive file.
ZDA	These files are the OpticStudio "session" files, which define the display configuration and open windows and settings for each saved ZMX lens file.
ZMF	ZMF are compilations of ZMX format files which are used to define lenses in the stock lens catalogs.
ZMX	ZMX files are the files used to store lens data. The ZMX file is a text file which contains the complete description of the lens, including apertures, wavelengths, prescription data, and the merit function.

ZPL	Zemax Programming Language macros.

General

These General settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab.
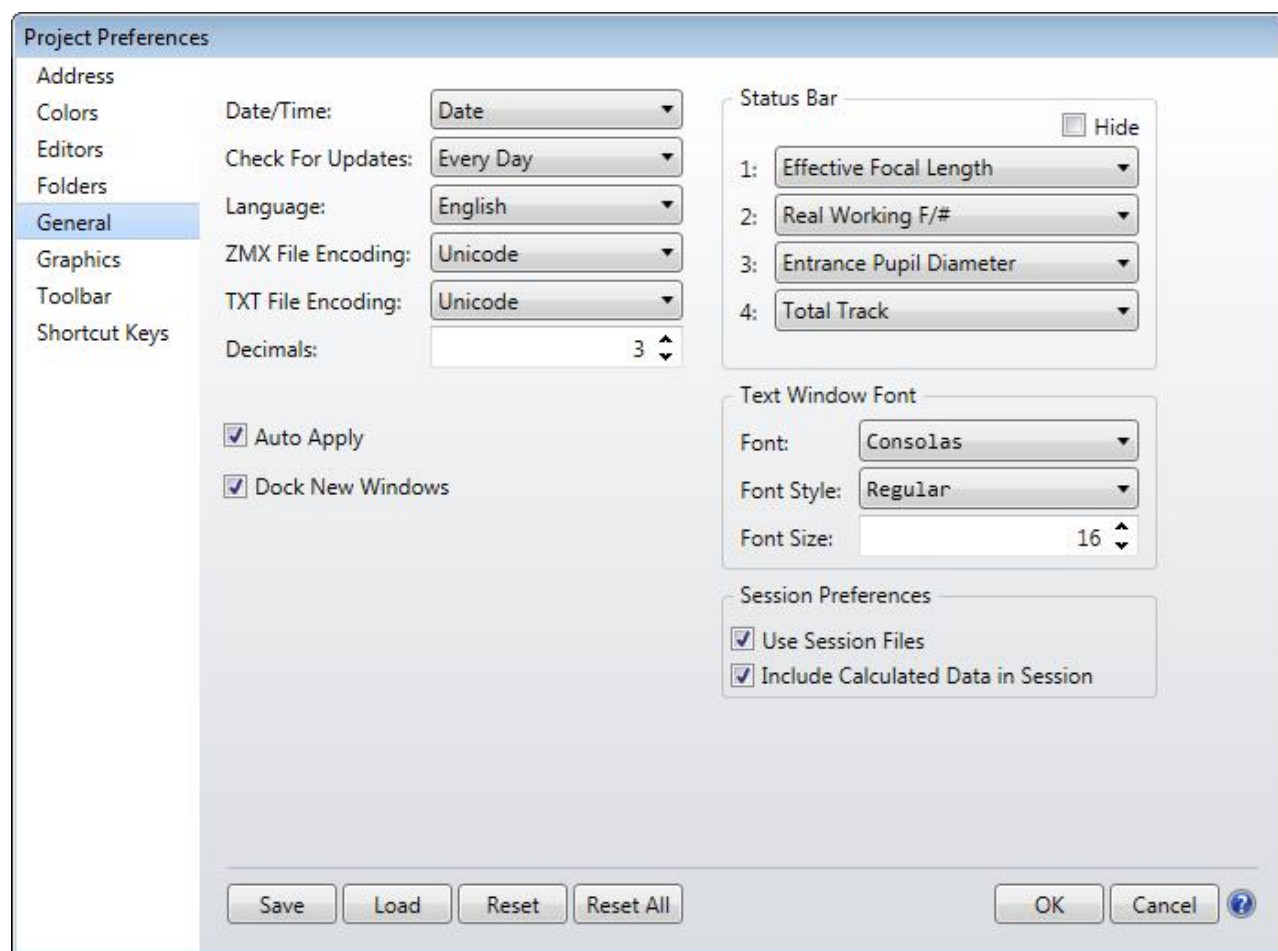
Settings:

Date/Time Select either None for no date or time, Date for the date only, and Date/Time for both the date and the time to be printed on plots and graphics.

Check For Updates Select how often Zemax should check to see if a new version of Zemax is available. Updates may be downloaded from www.zemax.com. An active Internet connection is required for Zemax to be able to check for updates. The "Check For Updates" button listed under the Help tab can be used to check for updates anytime.

Language Choose English, Japanese, Chinese, or Russian. This setting changes the user interface language of Zemax. Not all features fully support languages other than English.

ZMX File Encoding To support user interface languages other than English, Zemax by default uses Unicode file encoding for most files, including ZMX files. This control allows ZMX files to be written using ANSI (also called ASCII) encoding instead. The use of ANSI may allow older versions of Zemax to read ZMX files created by newer versions of Zemax. However, it is never advisable to use older versions of Zemax to read newer Zemax files as the newer version data format may not be fully compatible with older versions. Session and CFG files are usually not compatible with older versions of Zemax. ANSI encoding may only be used if the interface language is English.

TXT File Encoding This setting controls whether text data saved using the "Save Text" menu option (see "Text windows operations") uses ANSI or Unicode file encoding. See "ZMX File Encoding" for a discussion about Unicode. ANSI encoding may only be used if the interface language is English. Any file type not explicitly addressed here uses Unicode encoding unless otherwise stated. Zemax can convert any text file from Unicode to ANSI or vice versa using the CONVERTFILEFORMAT ZPL keyword. See "CONVERTFILEFORMAT".

Decimals This sets the number of decimal digits displayed in the System Explorer. Note that any trailing zeros are removed. For example, if the number of decimals is set to "3", and the value in the System Explorer is 0.55, then the value displayed will be 0.55 and not 0.550.

Auto Apply This determines the default setting (checked on, or off) for the Auto Apply option in the Settings of the various analysis windows.

Dock New Windows Toggles between docked and floating windows in the workspace.

Status Bar These settings determine which values are displayed in the Zemax status bar, or if the status bar is hidden. There are 4 regions which can be set to display various data, such as EFL, EPD, F/#, etc.

Text Window Font The font type, style, and size to be used for text based analysis windows, such as a ray fan text listing. To change the text displayed in the graphics based analysis windows see the "Graphics" section of the Project Preferences. To change the text displayed in the editors see the "Editors" section of the Project Preferences.

Use Session Files If checked, all open windows will be closed, and new windows will be opened and restored to their original locations when new lens files are loaded. If unchecked, files will be loaded without changing the arrangement of open windows.

Include Calculated Data in Session This determines the default setting for how data is saved in the session files. If checked, then calculation data for all open analysis windows (sequential mode) and/or all detectors (non-sequential mode) will be cached in the session file. This will allow the data to be restored when the lens file is loaded – significantly decreasing the time for file load – but also increases the size of the session file associated with the lens file

---

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Graphics

The Graphics settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab. These settings determine the size, color, and behavior of most Zemax graphics windows.
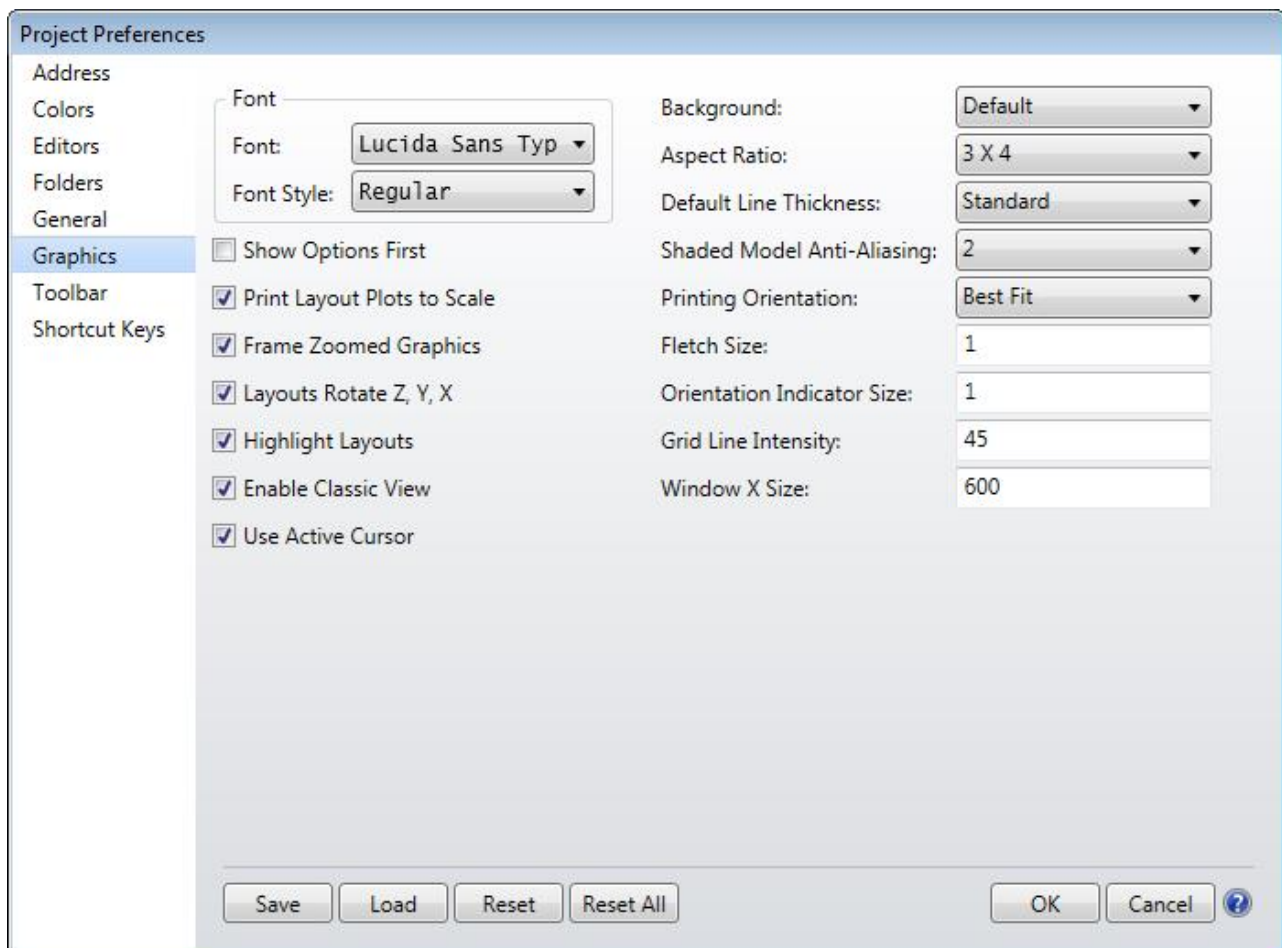
Settings

Font/Font Style Choose the font name and style for graphics.

Show Options First If this checkbox is selected, then the settings box will appear before any analysis graphic or text is computed and displayed.

Print Layout Plots to Scale Automatically size the printed graphic such that the output matches the true size of the system.

Frame Zoomed Graphics If selected, zoomed graphics which are pasted to the clipboard will be framed; otherwise, the frame will not be shown.

Layouts Rotate Z, Y, X If selected, all 3D Layout type plots will rotate the displayed lens first in Z, then Y, then X about the window's coordinates. This selection results in rotations that are similar to Zemax Coordinate Break conventions if the order flag is zero. If this option is not selected, then the displayed lens will be rotated first in X, then Y, then Z about the window's coordinates. This selection results in rotations that are similar to Zemax Coordinate Break conventions if the order flag is not zero.

Highlight Layouts If selected, then the current surface (in the LDE or EDE) or object (in the NSCE) will be highlighted in all open layout type plots. Moving the cursor from row to row will cause the layout plots to redraw with the selected surface or object highlighted. The highlight color is defined on the Colors tab. Highlighting does slow down navigation of the editors somewhat, because the graphics are continuously redrawn.

Enable Classic View This option enables an additional "Classic View" for some analysis windows. The example below shows the Classic View of the Physical Optics Propagation window with the Zemax sample file "Double Gauss 28 degree field.zmx".
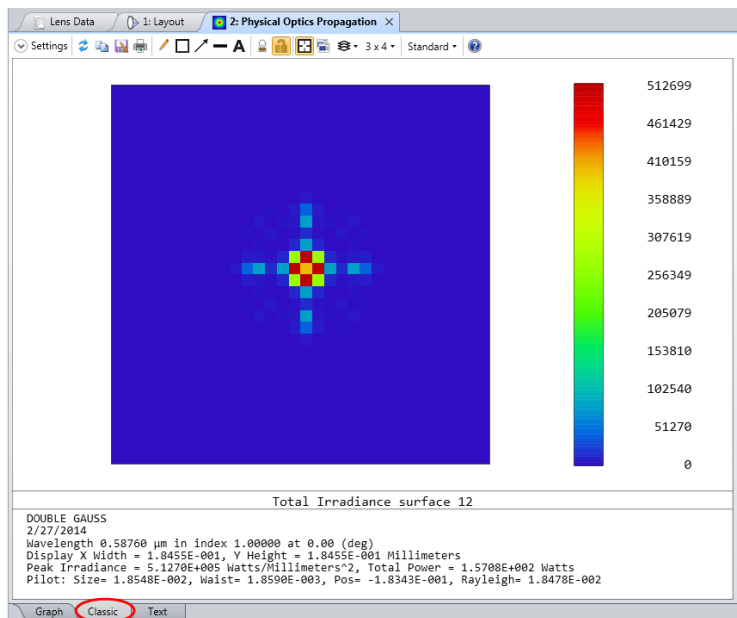
Use Active Cursor This setting determines whether or not the active cursor is initially on when creating a new window.

Background The background color on graphics windows can be selected from this drop down list.

Aspect Ratio The default aspect ratio for Zemax graphic windows is 3 x 4, which fits well on a standard 8.5 x 11 inch paper printer. For 11 x 17 inch printing, a 3 x 5 aspect ratio fits better. The 4 x 3 and 5 x 3 aspect ratios are taller than they are wide. This option selects the default aspect ratio for both printing and display graphics. Each graphic window may have its own aspect ratio, modified in its toolbar with a drop down setting.

Default Line Thickness Determines the default line thickness for the graphics in analysis windows.

Shaded Model Anti-Aliasing Controls the degree of anti-aliasing in shaded models, where higher values create a smoother image but may take a longer time to render

Print Orientation Controls whether the plot is printed with a Default, Best Fit, Portrait, or Landscape layout. "Default" uses the current printer settings. "Best Fit" uses the aspect ratio of the plot window and the printable area of the paper to determine whether portrait or landscape will result in a better fit.

Fletch Size Determines the relative size of fletch marks drawn on the various layout plots. The default value of 1.0 draws fletches at the default size; a value of 2.0 would draw the fletches twice as large as normal, and 0.5 would draw fletches at half size. Any value between 0.0 and 100.0 may be used.

Orientation Indicator Size Determines the relative size of the orientation indicator drawn on the various layout plots. The default value of 1.0 draws the indicator at the default size; a value of 2.0 would draw the indicator twice as large as normal, and 0.5 would draw the indicator at half size. Any value between 0.0 and 10.0 may be used. A value of zero will remove the indicator.

Grid Lines Intensity An integer value between 0 and 255 that indicates the intensity of the grey grid lines, if shown. A value of 0 will make lines that are pure white, while a value of 255 will make the grid lines pure black.

Window X Size The default x size for graphics and text windows in pixels. This can be adjusted to suit the monitor size and resolution.

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Toolbar

The Toolbar settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab. This allows the user to customize the icons after the separator in the Quick Access Toolbar.
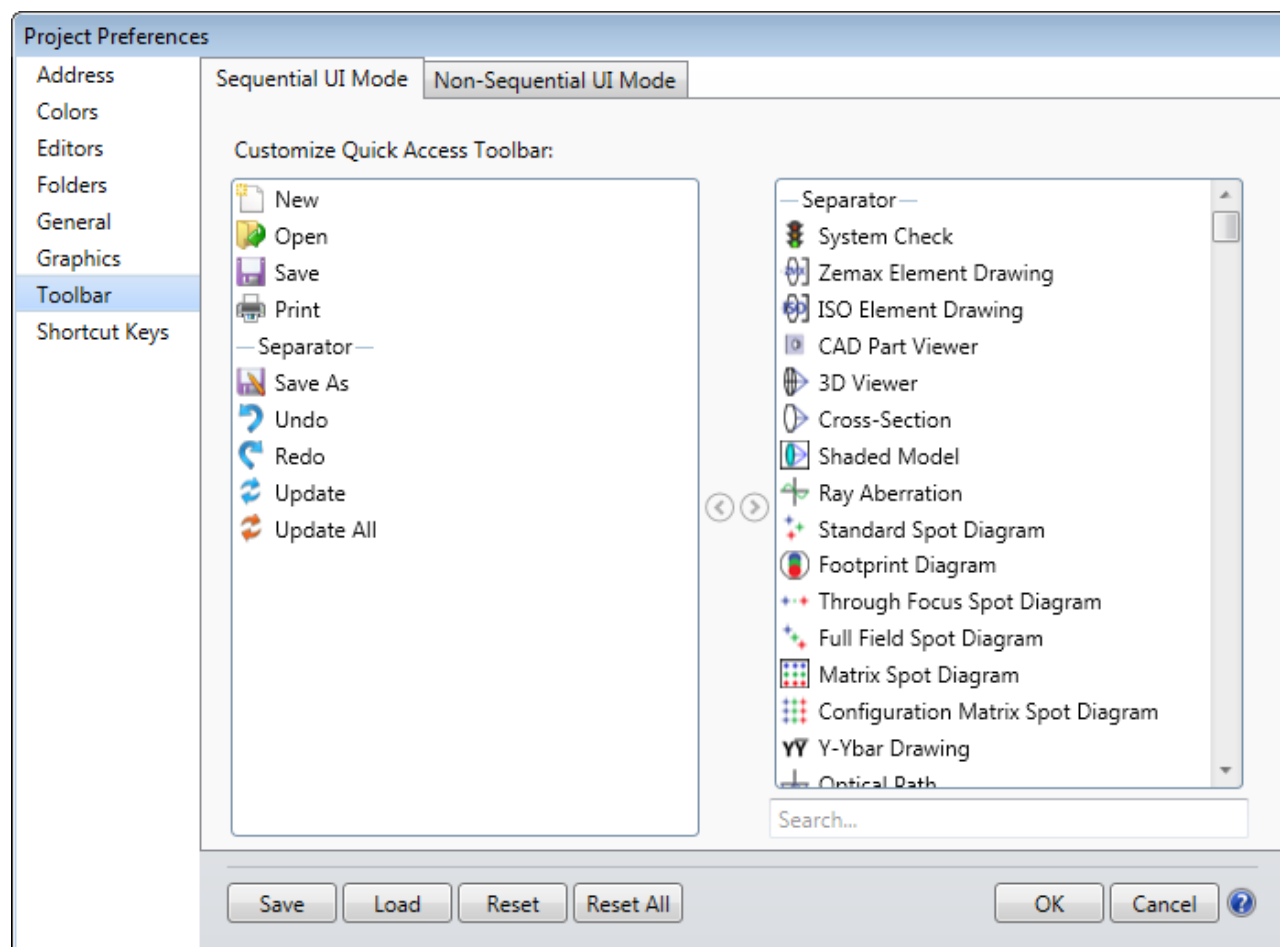
Move icons to and from the Quick Access Toolbar using the left and right arrows in the center. To add a new tool to the Quick Access Toolbar, first select the Sequential or Non-Sequential UI Mode tab, click on one of the features listed on the right, and then use the left arrow to move the feature to the toolbar.

For example, type "dock" to search for the "Dock All Windows" tool.

Click on the tool to select it, and then use the left arrow to move it to the toolbar:
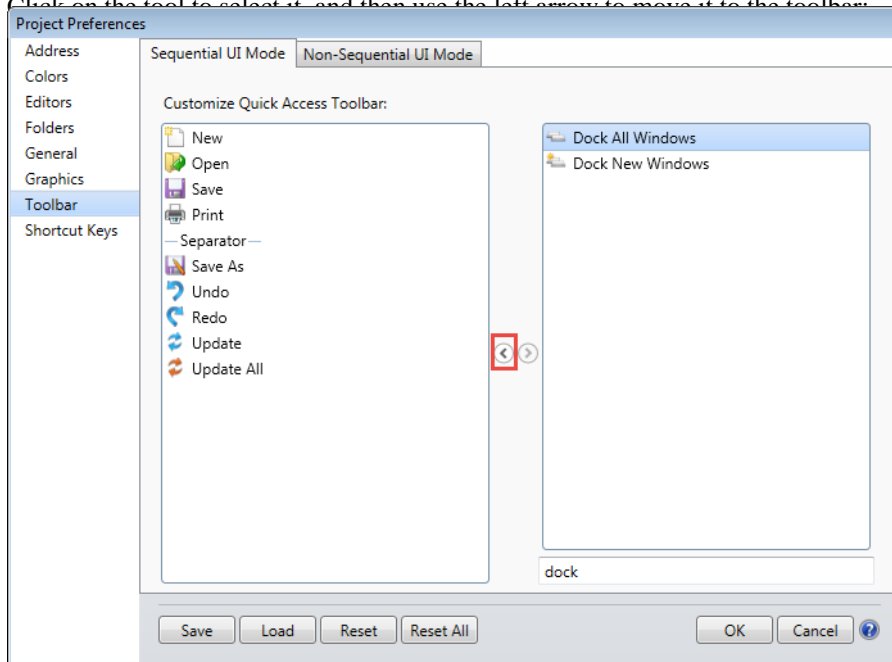
Similarly, the right arrow can be used to remove features from the toolbar.

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Shortcut Keys

The Shortcut Keys settings are available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab. This allows the user to look up and customize the shortcut keys.
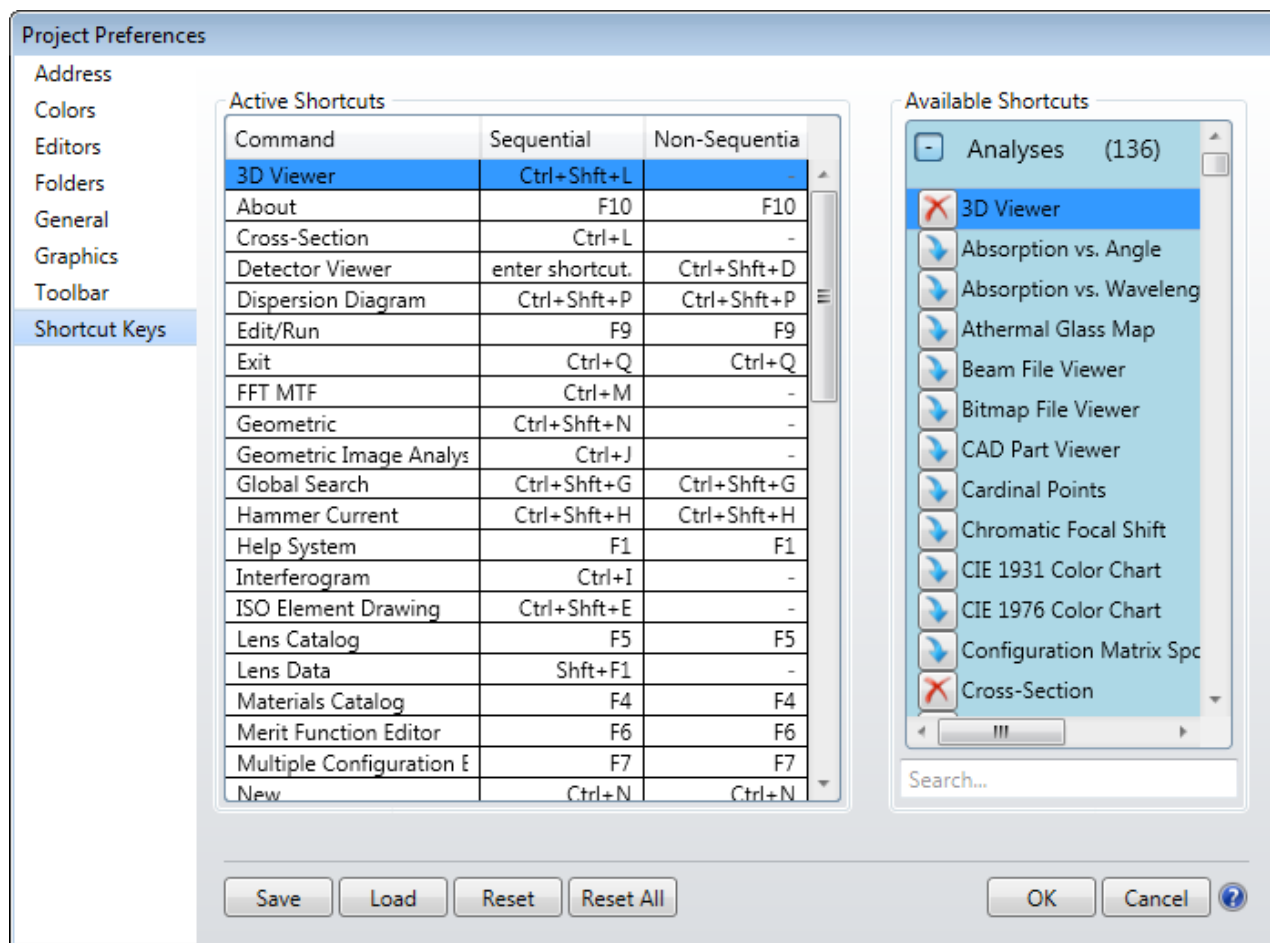
For a full list of the default shortcut key, please see theTecnhical Reference section called Shortcut Keys.

Active Shortcuts

The full list of active shortcuts.

To change a shortcut key, click on the Command's Sequential or Non-Sequential cell, and type the new combination of shortcut keys. For example, to change the sequential shortcut for the 3D Viewer to "F1", just press the "F1" key. To change it back to "Ctrl+Shft+L", type the keys in the correct order and press Enter. Note that the plus symbols are added automatically.

Available Shortcuts

Click on the add [icon] or remove [icon] button next to the name of the available shortcut to either add or remove it from the list of Active Shortcuts.

Button Functions

Save Saves the current settings to the OpticStudio configuration file.

Load Allows the user to load a previously saved configuration.

Reset/Reset All Resets one or all sections of the Project Preferences.

OK Closes the window and implements the changes without saving.

Cancel Closes the window without implementing any changes.

Scale Lens

The Scale Lens tool is available in the System section of the Setup tab. This feature will scale the entire lens by the specified factor. This is useful for scaling an existing design to a new focal length, for example. Wavelengths are not scaled. The scale lens feature may also be used to change the units from mm to inches, or other combinations of unit types.
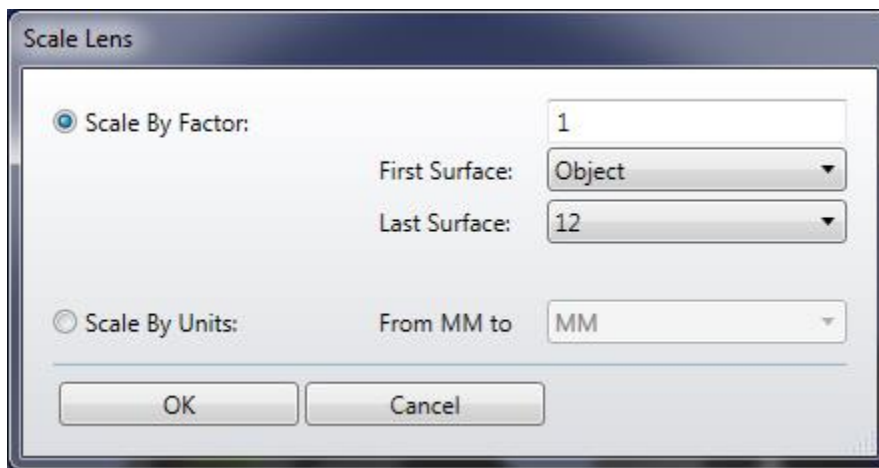
Settings:

Scale by factor If selected, then a scale factor may be entered directly.

First Surface The first surface of the range when scale by factor is selected.

Last Surface The last surface of the range when scale by factor is selected.

Scale by units If selected, then the lens will be converted by the selected units.

Scaling of data is performed based upon the units of measure for the data. If the scale factor is X, then data measured in lens units of length will be scaled by the factor X. Data measured in units of lens units squared (such as millimeters squared) will be scaled by X squared. Some polynomial coefficients, such as those on the Even Aspheric surface, have units that change from term to term, and Zemax accounts for this when scaling the data. Other parameters, such as the conic constant, are dimensionless, and are therefore not scaled.

Zemax will generally scale all lens data in the Lens Data Editor and the Non-Sequential Components Editor correctly. Zemax also will scale almost all of the data in the Merit Function Editor, the Multi- Configuration Editor, and the Tolerance Data Editor correctly as well.

Zemax will not attempt to scale some types of data, even though scaling would be appropriate for those types of data. These exceptions are for data items such as the multi-configuration operands PRAM and NPAR, that modify general parameter data of surfaces or objects, or the parameter data of user-defined DLL surfaces or objects. These data items

should be scaled manually, if required. In addition, Zemax will not attempt to scale the size of a CAD Part or CAD Assembly object. These objects should be scaled in the original creating program, if required.

Program Mode

The Program Mode buttons are available in the System section of the Setup tab. These buttons select the user interface and program mode.
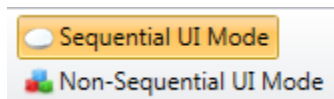
There are two distinct modes to the Zemax User Interface: Sequential or Mixed Sequential/Non-Sequential Design ("Sequential mode") Non-Sequential Design ("Non-Sequential mode" or "NSC mode"). The Sequential UI is recommended for the design, optimization and tolerancing of imaging systems. The Non-Sequential UI is recommended for illumination, multi-source systems, stray light, and general optical system modeling.

Sequential UI Mode

The Sequential UI Mode button is available in the System section of the Setup tab. This enables the sequential and hybrid non-sequential ray tracing user interface.

The following screenshot was taken with the Zemax sequential objectives sample file "Double Gauss 28 degree field.ZMX" in Sequential UI Mode:
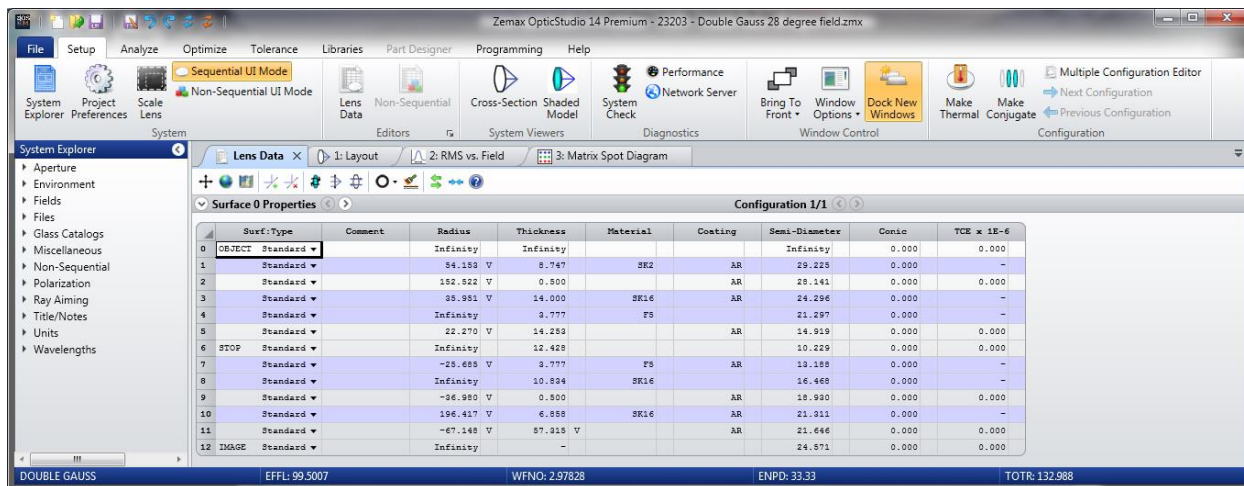
Most imaging systems are well described by optical surfaces which are sequential, which means that rays always trace from the object surface to surface 1, then 2, then 3, etc... in a strict sequence. Each ray "hits" each surface once and only

once in this predetermined sequence. Rays in sequential mode never trace from surface 3 to 9 then back to 1, for example. The sequential model is simple, numerically fast, and extremely useful and complete for many important cases.

In Sequential mode, all program features are available, including the ability to place a non-sequential group upon any surface. In Sequential mode, Zemax will trace rays which leave the object surface through all defined surfaces and non-sequential groups. This is the preferred mode for designing imaging systems, or any system requiring optimization, tolerancing, and detailed image analysis.

Non-Sequential UI Mode

The Non-Sequential UI Mode button is available in the System section of the Setup tab. This enables use of the pure non-sequential ray tracing interface.

The following screenshot was taken with the Zemax non-sequential scattering sample file "Fluorescence Example.ZMX" in Non-Sequential UI Mode:

Non-sequential means the rays trace in the actual physical order they hit various objects or surfaces, and not necessarily in the order the objects are listed in the software user interface. Note rays in a non-sequential trace may hit the same object repeatedly, and entirely miss other objects. Generally, the order in which objects are hit by rays depends upon the object geometry and the angle and position of the input ray. Objects which require or at least benefit from non-sequential ray tracing include prisms, light pipes, lens arrays, reflectors, and Fresnel lenses. Certain types of analysis, such as stray or scattered light effects, are only practical in a completely non-sequential environment.

In Non-Sequential mode, multiple changes are made to the user interface and to the lens data to simplify the analysis of these systems:

Only surface 1 is used, and all NSC objects are placed in a single non-sequential group placed on surface 1.

The object and image surfaces are not used.

The Lens Data Editor is not used. The Non-Sequential Components Editor (NSCE) is used instead as the primary editor.

The field dialog box is not used. The wavelengths dialog box is still used to define the wavelengths to be traced.

Only rays which originate from sources placed within the NSCE are traced.

Features which are not used in Non-Sequential mode are removed from the menus and button bars to simplify the user interface. These features include ray fans, MTF plots, spot diagrams, and many other features unique to sequential ray tracing. The primary means of analysis in Non-Sequential mode is tracing rays to one or more detector objects.

Switching UI Modes

Sequential to Non-Sequential

Switching from Sequential mode to Non-Sequential mode will cause all the sequential data to be deleted.

The following screenshot was taken with the Zemax sequential objectives sample file "Double Gauss 28 degree field.ZMX":
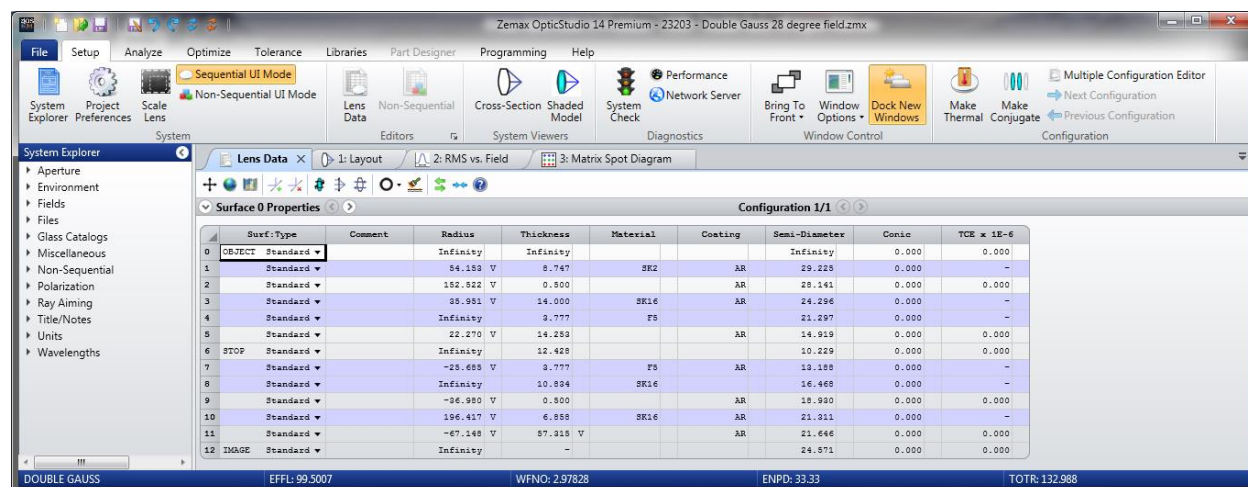
Select Non-Sequential UI Mode:
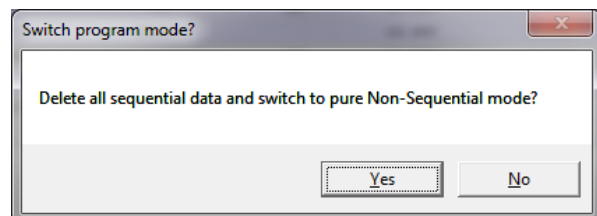
This changes the UI mode to Non-Sequential UI Mode, where the Non-Sequential Component Editor is available, but the Lens Data Editor is not:
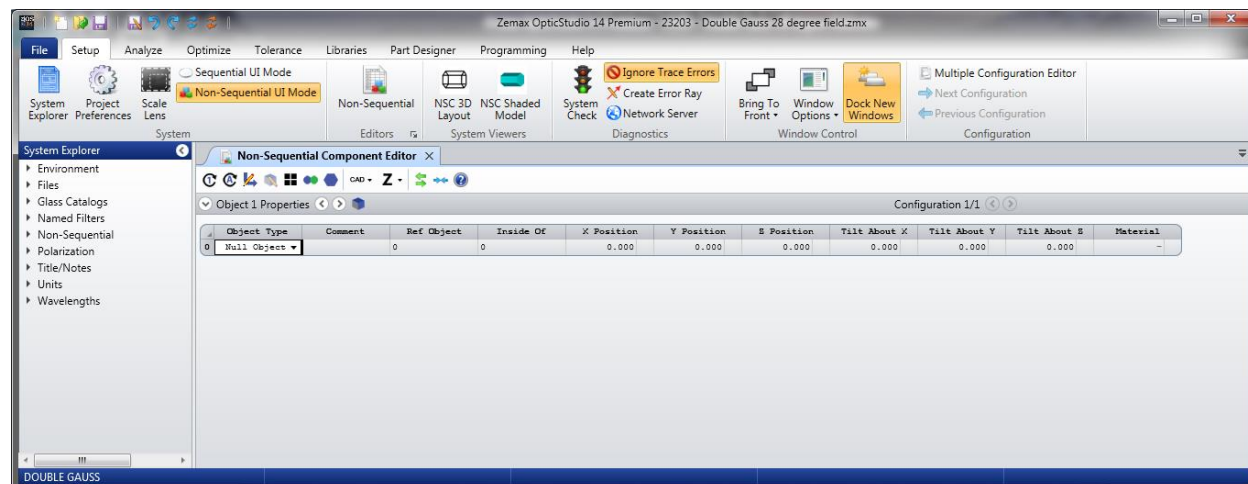
Non-Sequential mode to Sequential

It is always possible to switch from Non-Sequential mode to Sequential mode without loss of data.

The following screenshot was taken with the Zemax non-sequential scattering sample file "Fluorescence Example.ZMX":
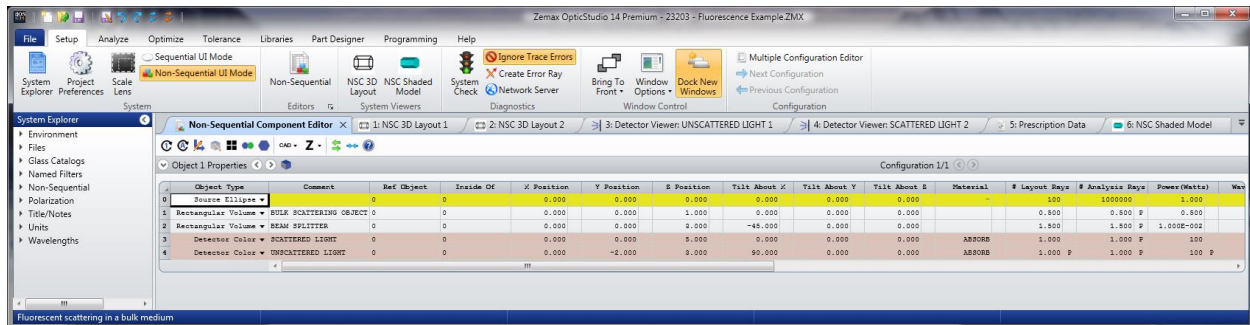
Select Sequential UI Mode:
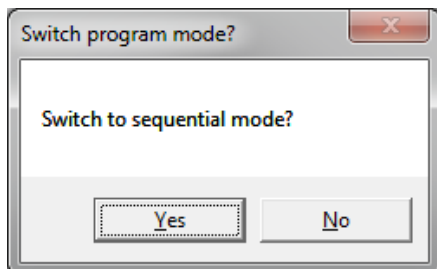
This changes the UI mode to Mixed Sequential/Non-Sequential UI Mode, where both the Lens Data Editor and the Non-Sequential Component Editor are available:
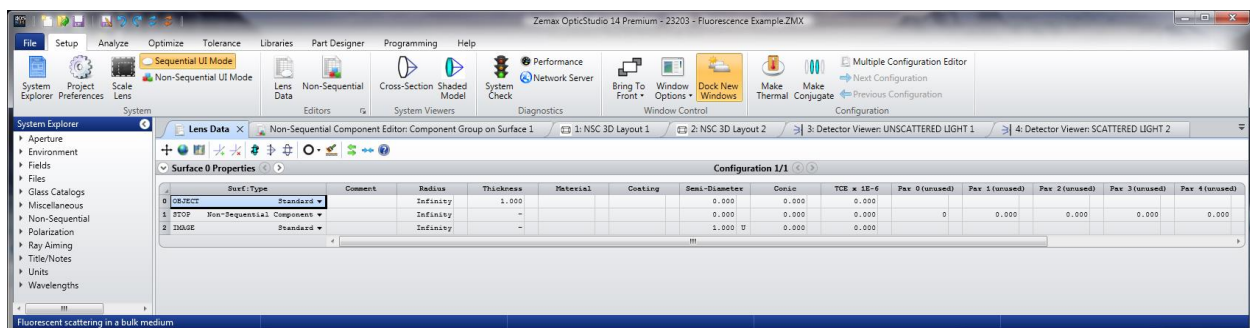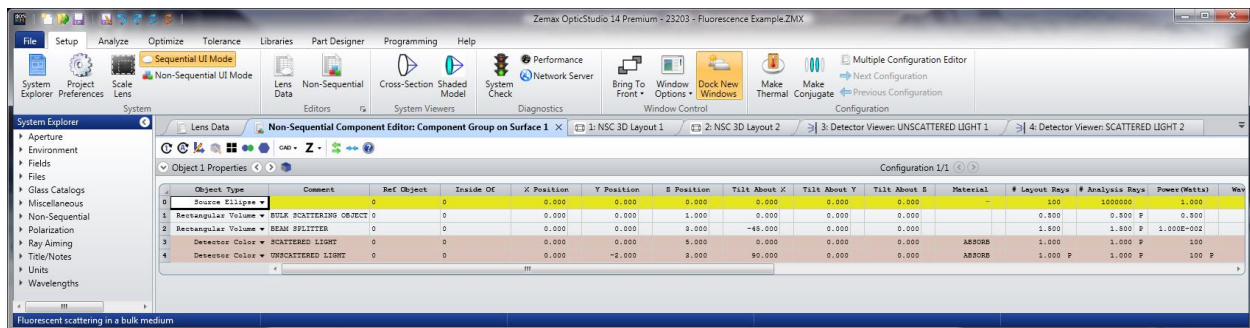
Editors Group

These are the Editors, available in the Setup tab.
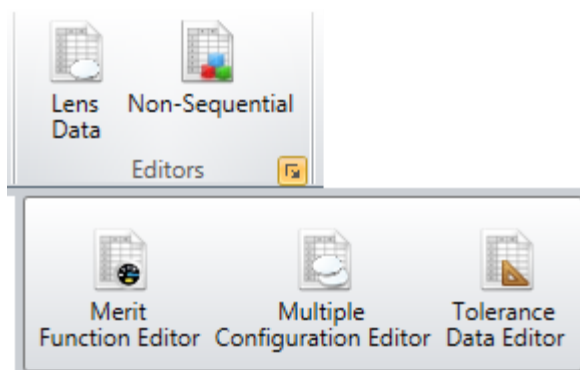
Lens Data Editor

The Lens Data Editor button is available in the Editors section of the Setup tab. The Lens Data Editor is the primary spreadsheet where the majority of the lens data will be entered. This data includes the radius of curvature, thickness, and glass for each surface in the system.

For a list of sequential surfaces, see [Sequential Surfaces](#).

The screenshots in this section were taken with the Zemax sequential objectives sample file "Double Gauss 28 degree field.ZMX".

Single lenses are defined by two surfaces (a front and a back), and the object and image also each require one surface. This fundamental data can be entered directly on the spreadsheet. When the Lens Data Editor is displayed, data can be entered on the spreadsheet by typing in the required values at the highlighted bar. Each column is labeled by the data type, and each row represents a single optical (or perhaps a dummy) surface. The cursor keys move the highlighted bar to whichever column is desired. Moving the cursor continuously to the right or left causes the display to scroll, which provides access to the other columns of data such as semi-diameters, conic constants, and "parameter" terms whose values depend upon the surface type. The display will scroll left to right or right to left. The page up and page down keys will move the bar to the top and bottom of the display. The display will also scroll up and down if required, when the number of lens surfaces is sufficiently large.

Note that initially (unless a lens has been loaded) three surfaces are shown; the object, stop, and image surfaces:
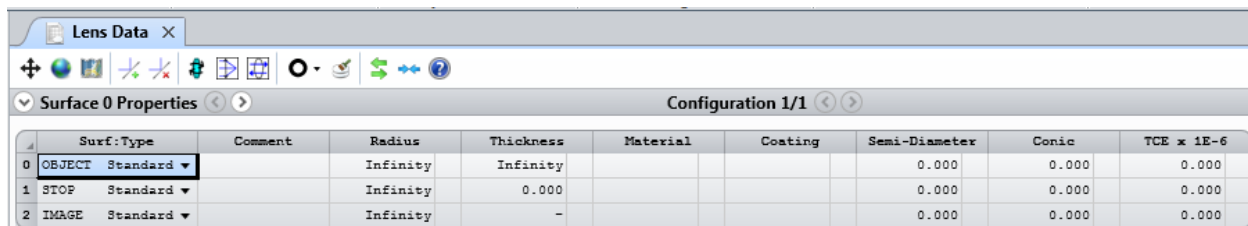
The object and image surfaces are permanent, and cannot be deleted. However, other surfaces can be inserted and deleted by using the Insert and Delete keys. No surface can be inserted in front of the object, or behind the image. In this context "in front of" means a smaller surface number, and "behind" means a larger surface number, in the sense that the light reaches the various surfaces sequentially. Zemax numbers surfaces from the object, being surface zero, upwards to the last surface, being the image surface.
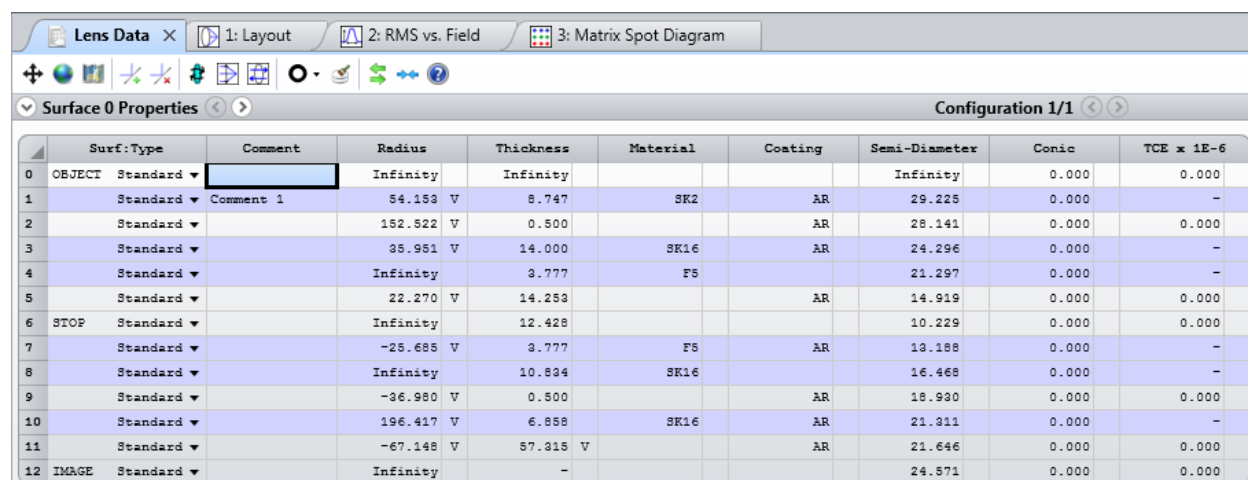
To enter values in the spreadsheet, move the cursor to the correct cell, and start typing. To edit values currently displayed, press the backspace key. Once you are editing the cell contents, you can use the left/right cursor keys and the

home and end keys to navigate around in the text. When the data has been changed, press any cursor key, or click on any other cell, or press enter.

There are also a few shortcuts available. To add a value to the current value, type a plus sign before the number. For example, if the displayed number is 10, typing "+5" and pressing enter will change it to 15. The symbols "*" and "/" also work. To subtract a value, use the minus sign followed by a space. For example entering "- 5" would change a 17 to a 12. Note the space between the "-" and the "5". If no space was entered, the program assumes you are entering a new value which happens to be negative. Entering "*-1" will change the sign of the number.

Data Columns

The following are data fields in the Lens Data Editor.

Surface Number & Type

The left most column of the Lens Data Editor displays the surface number and type for each surface. The surface number is zero for the object surface, 1 for the first surface, and so on to the image surface. The object surface is always surface zero, and the image surface is always the last surface, however the stop surface may be any surface number in between.

Some additional information is displayed adjacent to the surface number. When an aperture is defined on a surface, Zemax will display an asterisk "*" symbol next to the surface number. If surface tilt/decenter data is defined on a surface, the "+" symbol will be displayed. If both aperture and tilt/decenter data are defined the "#" symbol is displayed.

To define the surface type and other properties, see "The Surface Properties dialog box". To define the stop surface, see "Make Surface Stop". For information on defining surface aperture data, see "Aperture type and other aperture controls". For information on defining tilt/decenter data, see "Surface tilt/decenter tab".

Comment

Each surface has a comment field which can be used to enter text. The comment column is used for improving the readability of the lens prescription, and has no effect on ray tracing. Some analysis features also display the surface comments. The entire comment column can be hidden, see the "Options" menu description.

Radius

To enter or change the radius of curvature of a surface, move the cursor to the desired cell, and type in the new value. Radii data is always entered and displayed in lens units, which have dimensions of length.

Thickness

To enter or change the thickness of a surface, move the cursor to the desired cell, and type in the new value. Thickness data is always entered and displayed in lens units, which have dimensions of length. The thickness of a surface is the distance to the next surface. The only thickness that is not used is the thickness of the image surface.

Thicknesses always change sign after a mirror. After an odd number of mirrors, all thicknesses should be negative. This sign convention is independent of the number of mirrors, or the presence of coordinate breaks. This fundamental convention cannot be circumvented through the use of coordinate rotations of 180 degrees.

Material

The material used for each surface is usually specified by entering the name of the glass in the "Material" column on the Lens Data Editor. The glass name entered must be in one of the currently loaded glass catalogs. The default catalog is the "Schott" catalog; others are available. To use multiple glass catalogs, or to review, edit, or append the glass catalogs, see "Using Glass Catalogs". To specify that a particular surface be a mirror, use the glass name "mirror".

There is an optional "/P" command that may be appended to the glass name when entering a new glass. This option will cause Zemax to alter the curvatures before and/or after the surface to attempt to maintain a constant power for the lens element. For example, if the glass type is already BK7, entering a new glass type of "SF1/P" will change the glass type to SF1 while adjusting the radii of the surface and the surface after, to maintain constant power. Zemax considers the front and back vertex powers, as well as the element power correction due to the lens thickness. The algorithm adjusts both front and back curvatures if the lens is in air. If either the front or back surfaces are adjacent to another glass element, then only the curvature adjacent to the air is adjusted.

Semi-Diameter

The default semi-diameter is automatically calculated to be the radial clear aperture required to pass all rays from all field points. If any value is entered for the semi-diameter, then the new number will be retained and a "U" will be placed next to the value. This indicates the semi-diameter is user defined. For information on the use of semi-diameters, see "Semi-diameters". For information on vignetting rays with apertures, see the "Aperture" section of the "Surface properties".

Conic

Conic data is permitted on many different surface types. To enter or change the conic constant of a surface, move the cursor to the desired cell, and type in the new value. Conic constants are always dimensionless. See "Standard" for a discussion of how conic surfaces are defined.

TCE

The TCE value is the thermal coefficient of expansion. This is a dimensionless parameter. The value displayed or entered needs to be multiplied by 1e-6 to yield the actual value. Zemax currently uses the TCE value to model linear thermal expansion independent of the temperature range being used. The catalogs values supplied by the glass vendor are usually the TCE for the temperature range from -30 to +70 degrees Celsius, but Zemax will use the TCE value for any temperature range.

Once a material is defined in the glass catalog, Zemax uses the TCE specified for that material to determine the thermal expansion of the radius, center thickness, and other data for the surfaces using the material. However, if the material is not a solid, but is instead a gas or a liquid, then generally the thermal expansion is not governed by the material properties, but is instead determined by the edge thickness of the mounting material.

In this special case, Zemax needs to use the TCE supplied in the Lens Data Editor to define the mounting material properties, rather than the TCE supplied in the glass catalog. This can be accomplished by setting the "Ignore Thermal Expansion" switch for the material in the glass catalog.

Other Parameters

Parameter data consists of additional numeric values which define the properties of certain surface types. For more information on parameter data, See "Parameter data" in the "Sequential Surfaces" reference file.

Solve Types

Most data columns (such as radii and thickness) support one or more solves. Solves are functions which actively adjust specific values. Solves can be specified on radii, thicknesses, materials, semi-diameters, conics, parameters, and TCE values. To set a Solve Type on a cell, click on the small cell to the right of the data and select the solve type from the drop-down list. Some solves require additional parameters.

There are two available settings listed in the Solve Type drop down menu for all data columns:

Fixed Turns solves off.

Variable To make a parameter variable for optimization, click on the small cell to the right of parameter, and select Variable in the drop down Solve Type menu. Alternatively, click on the parameter and press Ctrl-Z on the keyboard. Ctrl-Z also removes the variable status, and acts like a toggle.

Suggestions for use

Solves are highly efficient, and should be used when possible instead of variables during optimization. For example, it is usually better to place a radius solve on the last surface to control effective focal length than to optimize for focal length explicitly. See "Optimization" for more information.

Summary of Solves

Solve type	Param 1	Param 2	Param 3	Param 4	Code	Integer
Curvature: Fixed						0
Curvature: Variable					V	1
Curvature: Marginal ray angle	Angle				M	2
Curvature: Chief ray angle	Angle				C	3
Curvature: Pickup	Surface	Scale factor		Column	P	4
Curvature: Marginal ray normal					N	5
Curvature: Chief ray normal					N	6
Curvature: Aplanatic					A	7
Curvature: Element power	Power				X	8
Curvature: Concentric with surface	Surface to be concentric to				S	9
Curvature: Concentric with radius	Surface to be concentric with				R	10
Curvature: F/#	Paraxial F/#				F	11
Curvature: ZPL Macro	Macro name				Z	12
Thickness: Fixed						0
Thickness: Variable					V	1
Thickness: Marginal ray height	Height	Pupil zone			M	2
Thickness: Chief ray height	Height				C	3
Thickness: Edge thickness	Thickness	Radial height (use zero for semi-diameter)			E	4

Thickness: Pickup	Surface	Scale factor	Offset	Column	P	5
Thickness: Optical path difference	OPD	Pupil zone			O	6
Thickness: Position	Surface	Length from surface			T	7
Thickness: Compensator	Surface	Sum of surface thicknesses			S	8
Thickness: Center of curvature	Surface to be at the COC of				X	9
Thickness: Pupil Position	-	-	-	-	U	10
Thickness: ZPL Macro	Macro name				Z	11
Glass: Fixed						0
Glass: Model	Index Nd	Abbe Vd	Dpgf			1
Glass: Pickup	Surface				P	2
Glass: Substitute	Catalog name				S	3
Glass: Offset	Index Nd offset	Abbe Vd offset			O	4
Semi-Diameter: Automatic						0
Semi-Diameter: Fixed					U	1
Semi-Diameter: Pickup	Surface	Scale factor		Column	P	2
Semi-Diameter: Maximum					M	3
Semi-Diameter: ZPL Macro	Macro name				Z	4
Conic: Fixed						0
Conic: Variable					V	1
Conic: Pickup	Surface	Scale factor		Column	P	2
Conic: ZPL Macro	Macro name				Z	3

Parameter: Fixed						0
Parameter: Variable					V	1
Parameter: Pickup	Surface	Scale factor	Offset	Column	P	2
Parameter: Chief ray	Field	Wavelength			C	3
Parameter: ZPL Macro	Macro name				Z	4
TCE: Fixed						0
TCE: Variable					V	1
TCE: Pickup	Surface				P	2

Radius Solves

Marginal Ray Angle

The effective focal length of the lens can be controlled by placing a paraxial marginal ray angle solve on the curvature of the last surface before the image. For example, suppose a lens has a 20 mm entrance pupil diameter, and it is desired to constrain the effective focal length to 100 mm. This requires a marginal ray exit angle from the last surface of -.1 (this number is 20 divided by 2 divided by 100, the minus sign is because the ray is converging, or going down toward the image surface). See the F number solve.

Chief Ray Angle

The chief ray angle solves work the same way as the marginal ray angle, except the paraxial chief ray is used for the calculations instead. The chief ray solve is useful for maintaining a particular magnification or for maintaining collimation. The letter "C" will appear in the curvature cell to indicate that a chief ray angle solve is active on that surface.

Pickup

The radius pickup solve uses a scaled value from another surface and column as the radius on the target surface. The pickup allows variables to be coupled together, and they will change under the influence of other solves, editing, or optimization. See also "Solve restrictions". To set a pickup solve from a ZPL macro, see "Integer codes for column numbers".

Marginal Ray Normal

This solve will force the surface to be normal to the paraxial marginal ray. This is also called an image-centered surface. These special surfaces introduce no spherical aberration or coma. The letter "N" will appear in the curvature cell to indicate that a normal ray angle solve is active on that surface.

Chief Ray Normal

This solve will force the surface to be normal to the paraxial chief ray. This is also called a pupil-centered surface. These special surfaces introduce no coma, astigmatism, or distortion. The letter "N" will appear in the curvature cell to indicate that a normal ray angle solve is active on that surface.

Aplanatic

This solve will force the surface to be aplanatic with respect to the paraxial marginal ray. These special surfaces introduce no spherical aberration, coma, or astigmatism. The letter "A" will appear in the curvature cell to indicate that an aplanatic solve is active on that surface.

Element Power

The power of an element is given by:

$$\varphi = c_1\,(n_2 - n_1) + c_2(n_3 - n_2) - c_1(n_2 - n_1) * c_2(n_3 - n_2) * \frac{t_2}{n_2}\,.$$

This solve adjusts the value of c_2, to maintain the specified element power. The solve is assumed to be on the second of two adjacent surfaces. This solve is ignored if the surface number is less than 2, or if n_3 and n_2 are identical at the primary wavelength.

Concentric Surf

This solve will force the curvature of the surface such that the surface is concentric about the specified surface. The specified surface must precede the surface on which the solve is placed. The letter "S" will appear in the cell to indicate that the surface concentric solve is active on that surface.

Concentric Radius

This solve will force the curvature of the surface such that the surface is concentric about the same point that the specified surface is concentric to. The specified surface must precede the surface on which the solve is placed. The letter "R" will appear in the cell to indicate that the radius concentric solve is active on that surface.

F Number

This solve will force the curvature of the surface such that the marginal ray angle exiting the surface is -1/2F where F is the paraxial F/#. The letter "F" will appear in the cell to indicate that the F/# solve is active on that surface.

ZPL Macro

For a complete description of ZPL Macro solves, see "Using ZPL Macro solves". Note curvature solves must return data in units of curvature, which is the reciprocal of the radius of curvature.

Thickness Solves
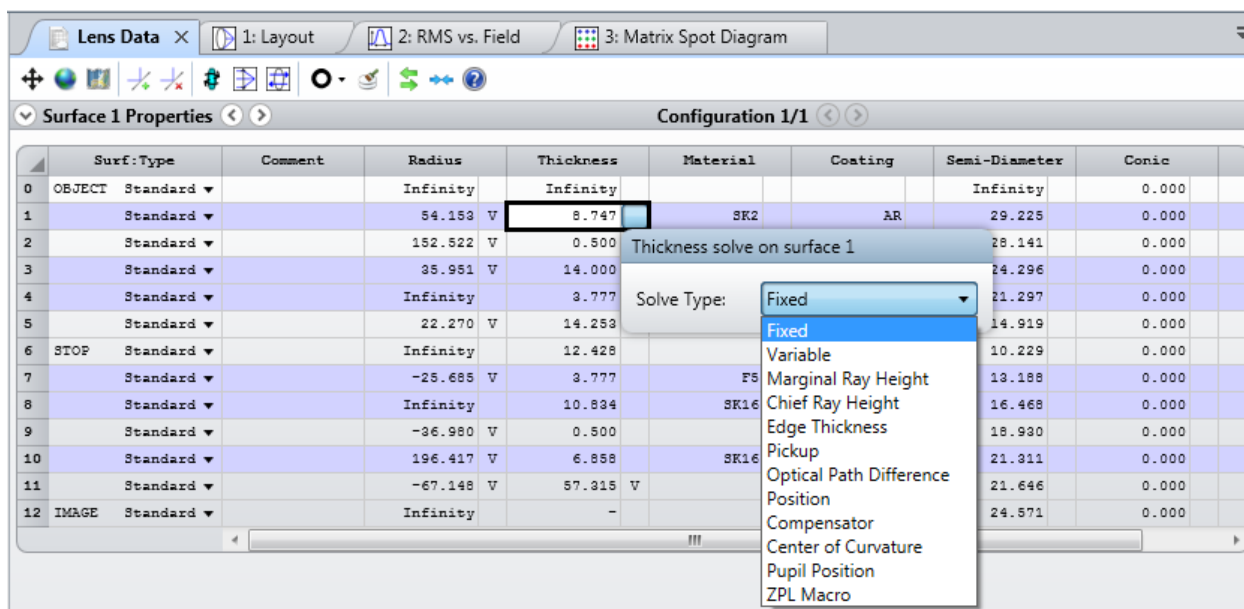
Marginal Ray Height

The most common thickness solve is the marginal ray height solve, which can be used to constrain the image surface to the paraxial focus. To move the image surface to the paraxial focus, click next to the thickness of the last surface before the image surface, and select "Marginal Ray Height" from the drop-down list of Solve Types. Since the marginal ray has a height of zero (it crosses the axis) at the paraxial image surface (assuming a rotationally symmetric system), we would like to set this last thickness to bring the image surface to the point where the marginal ray height is zero. The default height value is zero, which is desired, and so click on OK to exit the window, and the solve will adjust the thickness appropriately.

In principle, the solve could have been set for some other marginal ray height by entering a value other than zero for height, the first optional parameter. Note that any surface can have a marginal ray height solve, not just the thickness before the image surface. The "Height" is the height of the marginal ray on the tangent plane of the next surface (again, not necessarily the image surface). Note the height is measured at the point the ray intercepts the tangent plane, not the actual curved surface of the next surface.

The third value, "Pupil Zone", allows the ray pupil coordinate to be defined. The default is zero, which indicates that a paraxial ray should be used. Any non-zero value indicates that a real marginal ray is to be used. The zone value must be between -1 and 1. This is the Py coordinate, or the normalized entrance pupil coordinate in the y direction. This solve can be used to constrain that a particular ray, such as the .7 zonal ray, to have zero transverse aberration on axis. The letter "M" will appear in the thickness cell to indicate that a marginal ray height solve is active on that surface.

Chief Ray Height

This is similar to the marginal ray height solve, except the paraxial chief ray is used. This solve is useful for locating a surface at a pupil plane. The letter "C" will appear in the thickness cell to indicate that a chief ray height solve is active on that surface.

Edge Thickness

This solve dynamically adjusts the spacing between two surfaces to maintain a specified distance between the surfaces at a specified radial aperture. This is useful for preventing negative or overly sharp edges on elements. The letter "E" will appear in the thickness cell to indicate that an edge thickness solve is active on that surface. If the radial aperture is set to zero, then the current semi-diameter will be used. See "Edge thickness" for an important discussion on the definition of edge thickness.

Pickup

The thickness pickup solve uses a scaled and offset value from another surface and column as the thickness on the target surface. The pickup thickness T is given by: $T = O + S * V$, where V is the value of the source data, S is the scale factor, and O is the offset. See also "Solve restrictions". To set a pickup solve from a ZPL macro, see "Integer codes for column numbers".

Optical Path Difference

This solve will actively adjust the thickness to maintain a specific optical path difference at a specific pupil coordinate. The OPD is measured at the exit pupil, not at the surface that the solve is placed on. The two parameters to be set are the OPD in primary waves, and the pupil zone at which to evaluate the OPD. For example, to maintain the focal position so that the real marginal ray has the same optical path length as the real chief ray, define an OPD solve on the last thickness before the image surface. Set the OPD parameter to zero, and the pupil zone to 1.0. The letter "O" will appear to indicate the OPD solve is active. Perform an OPD plot and verify that the OPD is in fact zero at the pupil edge. Only the primary wavelength is used, and only the on-axis field is considered.

Position

The position solve maintains the "z" distance from a specified reference surface. If the reference surface precedes the surface on which the solve is placed, then the sum of the thicknesses from the reference surface to the surface following the solve surface will be maintained at the specified value. If the reference surface follows the surface on which the solve is placed, then the sum of the thicknesses from the solve surface to the reference surface will be maintained at the specified value. If the reference surface is the same as the solve surface, then the thickness of the solve surface will be set at the solve length value.

The position length solve is particularly useful for maintaining the length of a portion of a zoom lens to a fixed value. The solve can also be used to meet a total lens length constraint. In either case, the solve may eliminate optimization variables and operands, enhancing optimization convergence and speed. The letter "T" will appear in the thickness cell to indicate that a total length solve is active on that surface.

This solve assumes that all the surfaces in the affected range are in the same coordinate system.

Compensator

The thickness compensator solve is very similar to the position solve. This solve maintains the thickness on a surface so that the sum of the surface thickness and the thickness on another "reference" surface equals a constant. In equation form, this solve maintains the condition $T = S - R$, where S is the sum of the thicknesses of the two surfaces, and R is the thickness of the reference surface. The reference surface must precede the surface with the solve.

Center of Curvature

This solve will adjust the thickness of the surface to place the following surface at the center of curvature of any prior surface. The letter "X" will appear in the cell to indicate that the center of curvature solve is active on that surface.

Pupil Position

This solve will place the next surface at the pupil position for the optical space following refraction from the current surface. The pupil position is determined by tracing real, differential rays about the central field chief ray.

ZPL Macro

For a complete description of ZPL Macro solves, see "Using ZPL Macro solves".

Material Solves
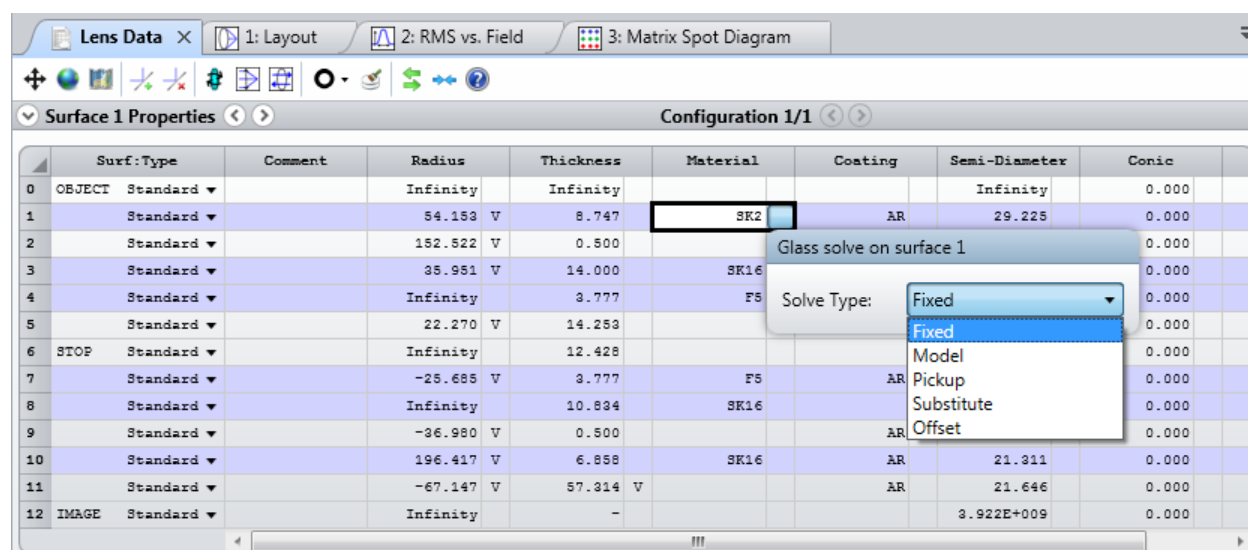
Model

The "Model" Material Solve is more like the "Variable" Radius/Thickness setting than a true solve. Zemax will come up with a "model" glass by varying the index, Abbe number, and partial dispersion term at d-light to idealize dispersion of the glass. Using the Ctrl-Z shortcut on the Material column will also set the solve to "Model". The model glass data values can then be optimized using the standard optimization methods.

Unconstrained glass optimization usually will lead to very high index materials being selected. This is because surfaces with high refractivity (a large difference in index across the boundary) need less curvature than low refractivity surfaces to have the same optical power. Lower curvature surfaces introduce less aberration.

Unfortunately, high index materials are expensive, heavy, harder to fabricate, and may be brittle, delicate, or susceptible to stains and scratches. Also, very high index materials do not always exist; there are few glasses (for the visible spectrum) available with an index higher than about 1.9. The Abbe number also is limited to the range of roughly 20 to 80. Therefore, it is essential to limit the index and Abbe numbers to reasonable ranges during optimization. The partial dispersion deviation also must be limited in range.

Even with reasonable limitations, one disadvantage of the model glass method is that the optimized parameters and resulting index values may not correspond to any physically existing glass. Another disadvantage is that model glasses are only sufficiently accurate in the visible spectrum because the dispersion is idealized at d-light (.5875618 micrometers). Outside the visible wavelength range, such as in the ultraviolet or infrared, the model glass is not accurate and should not be used.

For more information on model glasses, see "Using model glasses".

Pickup

The glass pickup solve uses a value from another surface as the glass on the target surface. See also "Solve restrictions".

Substitute

Like the "Model" setting, the "Substitute" setting is not a true solve, and is similar to the "Variable" Radius/Thickness setting. If the glass Solve Type is set to "Substitute", then the global optimization algorithms are permitted to change the glass type during optimization.

If no catalog name is given (i.e. the catalog field is blank) then glasses may be selected from all catalogs selected for use in the Glass Catalogs section of the System Explorer. If a catalog name is given (i.e. Hoya) then only glasses from that one catalog will be selected.

To prevent only certain glasses from being chosen during optimization, choose "Exclude Substitution" for the glass that should be avoided on the glass catalog dialog box, or use a glass substitution template; see "Using glass substitution" for more information.

Offset

The offset solve allows a small change in index and/or Abbe number to be added to the index as computed by the dispersion formulas and the glass catalog dispersion data. The primary use of this solve is for tolerancing. There are two conditions for using the index and Abbe numbers in computing the change in index as a function of wavelength:

The minimum wavelength in use is greater than 0.3 micrometers and less than 2.5 micrometers

The dispersion data in the glass catalog must span the wavelength range of 0.4861327 to 0.6562725 micrometers.

If these conditions are met, then the change in index is given by the difference in the model glass index computed at the base index (Nd) and Abbe number (Vd) as compared to the offset Nd and Vd values. See the discussion of the "Model" glass for a discussion of the model glass properties. In equation form the index is:

$$n = n_{base} + n(N_d + \Delta N_d, V_d + \Delta V_d, P_d) - n(N_d, V_d, P_d)$$

where the function n() is the model glass function. Note this model will add a different index offset to the index at each wavelength. If the wavelength band criteria listed above is not met, then the index offset is added to all the base index values:

$$n = n_{base} + \Delta N_d$$

The Abbe offset is ignored in this case.

Semi-Diameter Solves
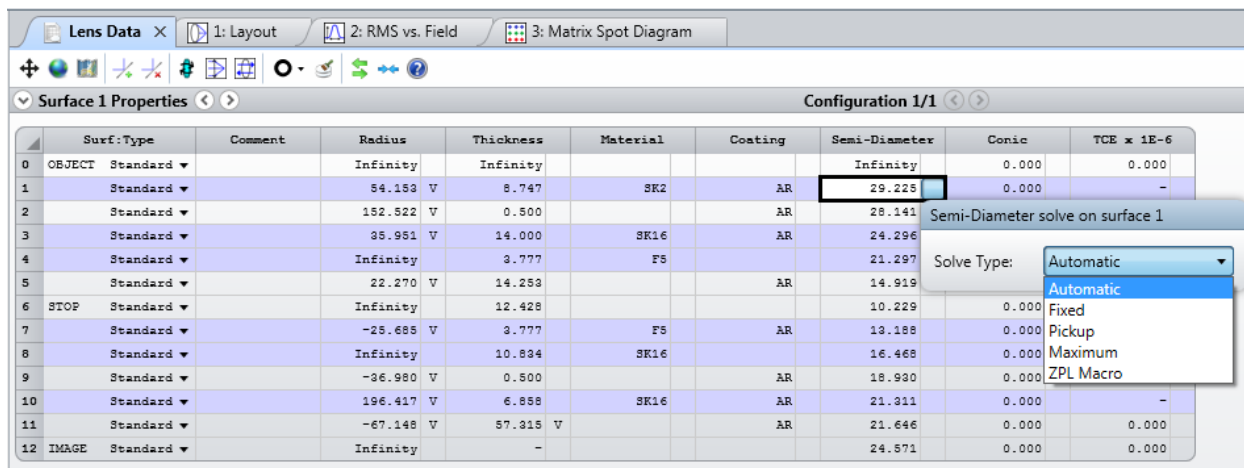
Automatic

The default semi-diameter is automatically calculated to be the radial clear aperture required to pass all rays from all field points

Pickup

The semi-diameter pickup solve uses a scaled value from another surface and column as the semi-diameter on the target surface. See also "Solve restrictions". To set a pickup solve from a ZPL macro, see "Integer codes for column numbers".

Maximum

The maximum solve is used for setting the semi-diameter value to the maximum required across multiple configurations. For example, a zoom lens which has three configurations will generally have three different semi-diameter values for every surface that is using "automatic" semi-diameters. The maximum solve will compute the semi-diameter for each configuration, and then use the largest of the values.

ZPL Macro

For a complete description of ZPL Macro solves, see "Using ZPL Macro solves".

Conic Solves
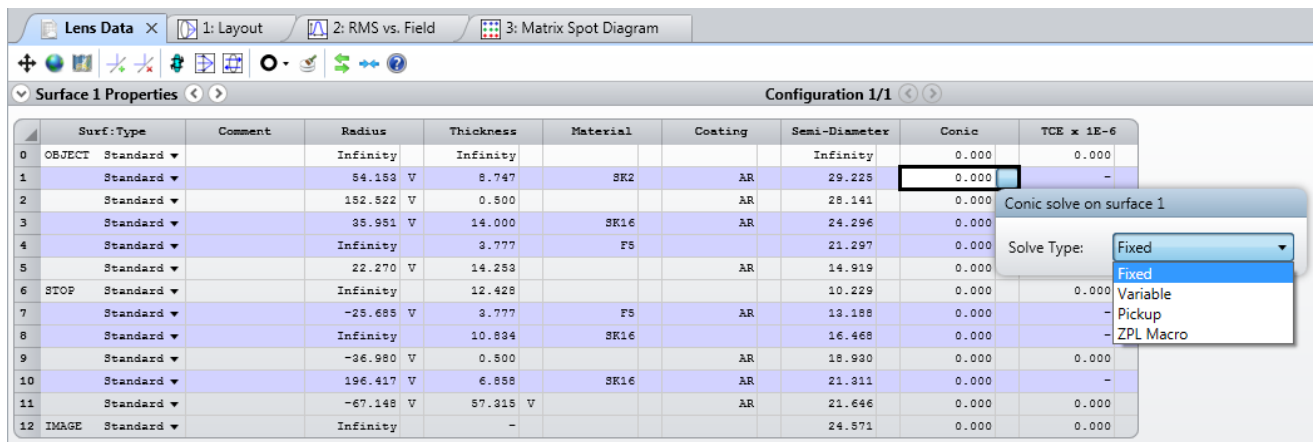
Pickup

The conic pickup solve uses a scaled value from another surface and column as the conic on the target surface. See also "Solve restrictions". To set a pickup solve from a ZPL macro, see "Integer codes for column numbers".

ZPL Macro

For a complete description of ZPL Macro solves, see "Using ZPL Macro solves".

TCE Solve
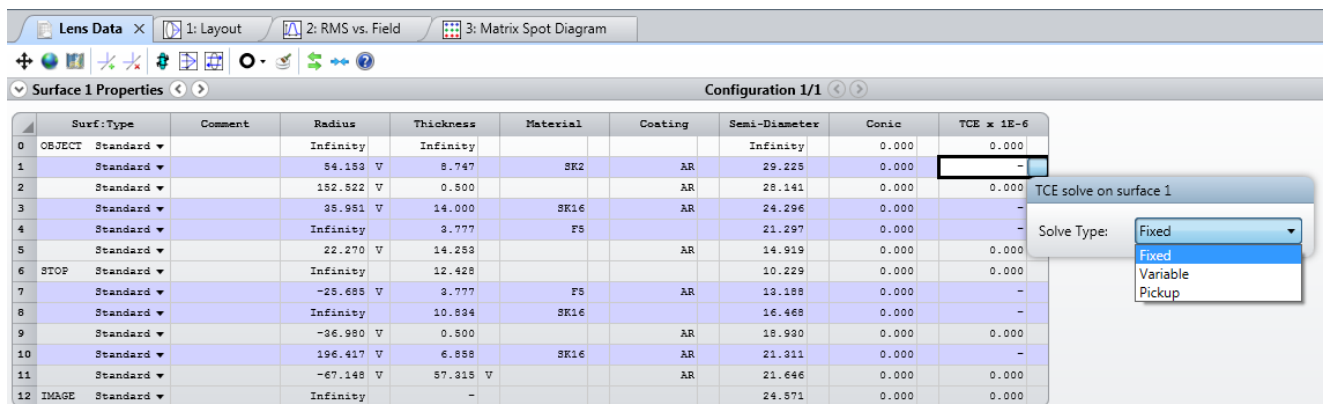
Pickup

The TCE pickup solve uses a value from another surface as the TCE on the target surface. See also "Solve restrictions".

Parameter Solves
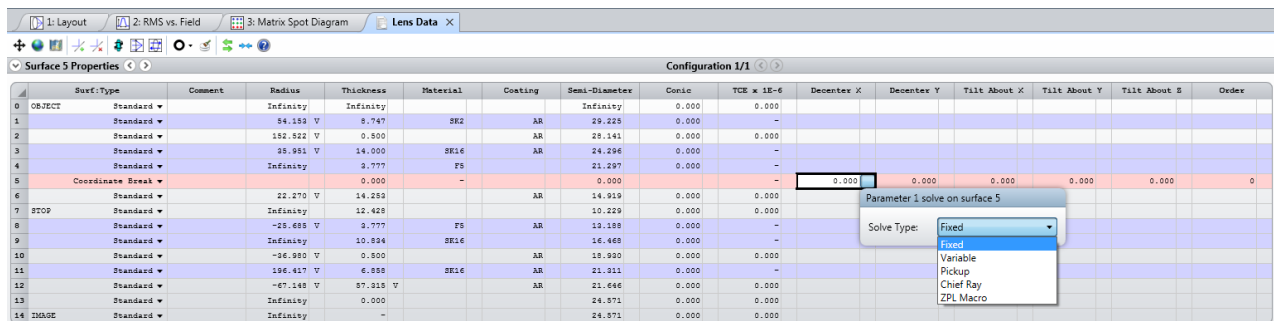
Pickup

The Parameter Pickup solve uses a scaled and offset value from another surface and column as the parameter on the target surface. The pickup parameter is given by: $P' = O + S * V$, where V is the value of the source data, S is the scale factor, and O is the offset. See also "Solve restrictions". To set a pickup solve from a ZPL macro, see "Integer codes for column numbers".

Chief Ray

The Chief Ray solve works only on coordinate break surfaces, and only on the first four parameters. When placed on a decenter x or y parameter, the solve will set the decenter to place the selected wavelength (use zero for primary wavelength) real chief ray from the selected field position at an x or y coordinate of zero on the coordinate break surface, respectively. If set on the tilt x and tilt y parameters, the tilt angles are adjusted to make the exiting chief ray angle zero in the y and x directions, respectively. The order flag on the coordinate break surface will determine the order in which the parameters are fit, and the solution may not be unique. The output chief ray coordinates can be viewed in the Ray Trace Data window (from the Rays & Spots button in the Image Quality section of the Analyze tab); and it is a good idea to verify the solve works correctly. This solve is not allowed when the stop surface follows the surface the solve is placed on and ray aiming (see "Ray Aiming") is on.

There are times when the resulting values will yield an output chief ray angle or coordinate that is small, but not zero. This is caused by the non-orthogonal rotations not always being able to completely "undo" an arbitrary rotation in a single set of matrix rotations. There are two solutions to this problem:

1) Try changing the "order" flag from 1 to 0 or 0 to 1, then update the system, to see which set of coordinate transforms works better.

2) Use two adjacent coordinate break surfaces with identical chief ray solves. This is easy to do by copying and pasting the entire coordinate break surface. The second coordinate break will usually bring the chief ray many orders of magnitude closer to the desired zero coordinates or angles.

ZPL Macro

For a complete description of ZPL Macro solves, see "Using ZPL Macro solves".

Restrictions

Solves are evaluated sequentially from the first surface to the image surface, in this order: curvature, thickness, glass, semi-diameter, conic, parameters (in order left to right), TCE, and extra data values (in order left to right). Because of this order, pickup solves may only pickup values from prior surfaces, or from the same surface if the source column precedes the target column.

Because solves can affect the entrance pupil position, Zemax does not allow certain types of solves, such as marginal ray height solves or macro solves, to be placed before the stop surface. This will avoid ambiguous or incorrect settings for the solved parameters. There is no way to predict the results of solves placed before the stop. A similar problem arises when a marginal ray solve is used on the last surface to control the focal length - the aperture definition must be entrance pupil diameter (rather than F/#) to keep the solution unique.

Lens Data Editor Toolbar

The Lens Data Editor Toolbar is available at the top of the Lens Data Editor window. This provides quick access to tools for the Lens Data Editor.
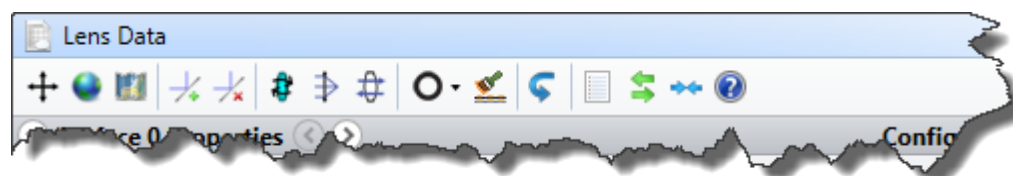
The buttons in the Lens Data Editor Toolbar are listed below.

Tilt/Decenter Elements

Add tilt and decenter to the specific range of surfaces.

The feature inserts coordinate break and dummy surfaces as required to tilt and/or decenter a single surface or a range of surfaces. For more information on coordinate breaks see "Coordinate Break".
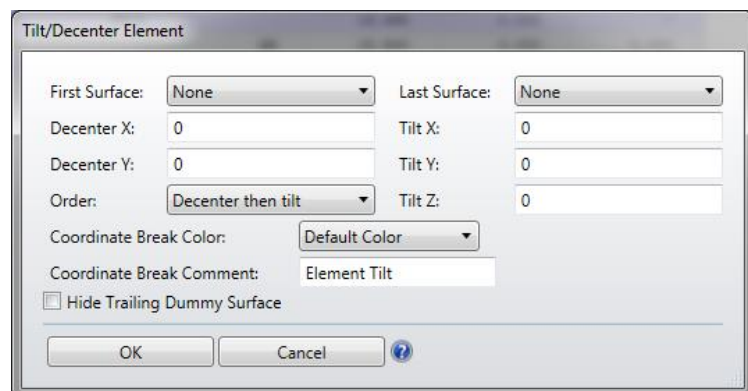
Settings:

First Surface This is the first surface of the lens group to be tilted/decentered.

Last Surface The last surface of the lens group to be tilted/decentered.

Decenter X, Y The X and Y decenters in lens units.

Tilt X, Y, Z The rotation about X, Y, and Z in degrees.

Order Either tilt then decenter, or decenter then tilt.

Coordinate Break Color Selects the spreadsheet display row color for the inserted coordinate breaks. See "Row Color" for a description of Row color.

Coordinate Break Comment The comment string for the inserted coordinate break surfaces.

Hide Trailing Dummy Surface If checked, any trailing dummy surface inserted as part of the tilt and decenter tool modifications will be checked with the options "Skip Rays To This Surface" and "Do Not Draw This Surface". See the "Draw" section of the "Surface Properties".

Local To Global

Switch from local to global coordinate system.

This feature combines, modifies, and deletes existing coordinate break surfaces as required to position groups of surfaces using a single decenter and rotation relative to a reference surface. The original position of the surfaces is not modified, only the coordinate decenters and rotations are modified. See also "Global To Local".

Settings:

First/Last Surface The range of surfaces to modify. The first surface must precede the last surface. Neither of these surfaces may be a coordinate break.

Reference Surface The surface number to which the coordinates and rotations are referenced. This surface must precede the First Surface.

This feature automates the process of placing surfaces in a common reference coordinate system. A group of surfaces is defined by any number of surfaces bounded by a coordinate break or the image surface. Each group is then positioned relative to the reference surface by three coordinate break surfaces. The first coordinate break returns the coordinate system to the reference surface. The second coordinate break positions the group using the decenter x, decenter y, and thickness (z coordinate) relative to the reference surface. Finally, the third coordinate break rotates the group into the correct orientation. Each set of three coordinate breaks precedes each group of surfaces in the system, within the defined surface range. The primary application of this tool is to place all groups of surfaces relative to a common reference for tolerancing or perturbation analysis.

This feature is not supported if any surface thickness, ignored surfaces, or any coordinate break properties within the range are controlled by the multi-configuration editor.

Global To Local

Switch from global to local coordinate system.

This feature combines all consecutive coordinate breaks into single coordinate breaks. When multiple coordinate breaks are placed on consecutive surfaces, they can be combined into a single coordinate break with the same result. The original position of the surfaces is not modified, only the coordinate decenters and rotations are modified. Each group of surfaces will be referenced to the previous group's local coordinates. The combination of consecutive coordinate breaks can also be used to reverse the process performed by the Local To Global tool. See also "Local To Global".

This feature is not supported if any coordinate break properties within the range are controlled by the multi-configuration editor.

Settings:

First/Last Surface The range of surfaces to modify. The first surface must precede the last surface. Neither of these surfaces may be a coordinate break.

Order Select "Forward" to use coordinate breaks that decenter then tilt, or "Reverse" to use coordinate breaks that tilt and then decenter. For more information, see "The order flag".

Add Fold Mirror

Make the specified surface a fold mirror and inserts the required coordinate breaks to bend a beam.

This feature inserts two dummy surfaces, one before and one after the selected fold surface. The fold surface then becomes a mirror, and the two newly inserted adjacent surfaces are set to be coordinate breaks with the appropriate tilt angles. The second tilt angle is set as a pickup from the first tilt angle. Finally, all subsequent surface thicknesses and curvatures are changed sign to account for the new mirror.

This feature may not provide useful or even correct alterations if the selected fold surface is not a plane, standard type dummy surface in air. The dummy surface should be located at the desired fold mirror position before using this feature. For example, to insert a fold mirror midway between two lenses 100 mm apart, a dummy surface should be inserted between the two lenses, with the thickness before and after the dummy being set to 50 mm. The dummy surface then should be used as the fold surface.

This feature may not work properly for lenses with multi-configuration operands that alter any parameter or glass data for any surfaces from the fold surface onward.
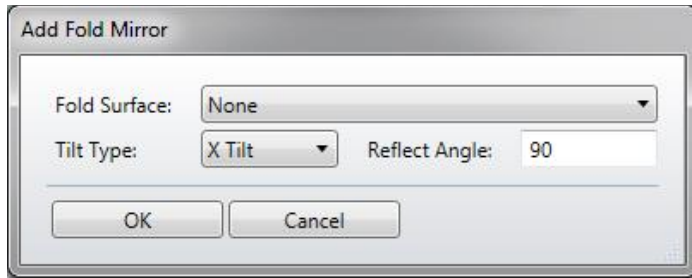
Settings:

Fold surface Selects which surface will become the fold mirror. The selected surface should be a dummy surface already located at the desired fold position.

Reflect angle The angle between the incident and reflected beam.

Tilt type Select either tilt about the local x or y axis.

Limitations of reversing surfaces following the fold mirror

After inserting a mirror, all subsequent surfaces must be reversed in the sense that +z becomes -z for the rays to trace the same. For most Zemax surfaces, this means changing the sign of the radius of curvature, or any other parameters that determine the sign of the surface sag. For user defined surfaces, Zemax cannot automatically make this adjustment. For non-sequential component (NSC) surfaces, it may not be possible. Zemax attempts to reverse NSC surfaces by changing the sign of the Z coordinate of the exit port, and rotating all components around the y axis 180 degrees. However, this method fails if the objects inside the NSC are not symmetric in their local XZ and YZ planes. Because the z parity of the objects must be modified, there is no general way to do this for all NSC objects.

Delete Fold Mirror

Deletes an existing fold mirror, including any neighboring coordinate breaks.

This feature deletes a fold mirror. If the mirror surface is preceded by a coordinate break with zero thickness, the coordinate break will be deleted. If the mirror surface is followed by a coordinate break, and the mirror surface has zero thickness, the following coordinate break will be deleted. The thicknesses of all deleted surfaces will be added to preceding surfaces as required. All following surfaces and thicknesses will be automatically reversed as required to follow the correct sign convention. See "Limitations of reversing surfaces following the fold mirror" for details. This feature may not provide useful or even correct alterations if the selected surface is not a plane, standard type mirror surface in air.
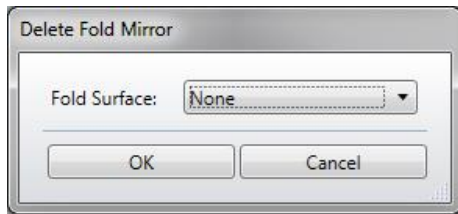
Settings:

Fold surface Selects which mirror surface will be deleted.

Reverse Elements

Reverses a lens element or group.

The feature may not work correctly in all cases if mirrors, coordinate breaks, multi-configuration controlled data, surface tilts and decenters, or non-standard surfaces are included in the range of surfaces. Solves on data affected by the reversal are removed by this feature. Check the reversed system results carefully to verify that the desired result was achieved.
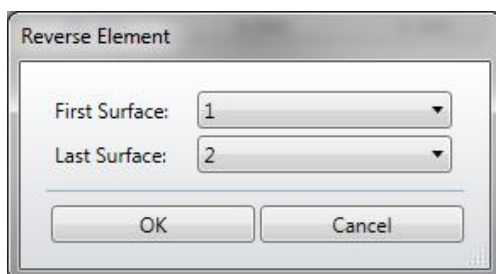
Settings:

First Surface This is the first surface of the lens group to be reversed.

Last Surface The last surface of the lens group to be reversed.

Make Focal

Scale the lens to the desired focal length.
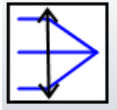
Make focal length is similar to scale lens, except the desired focal length is typed in directly. The entire lens is then scaled to make the focal length the specified value.
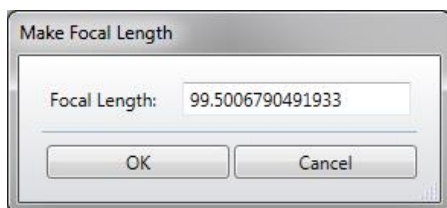
Make Double Pass

Create a second pass through the specified surfaces that represents a reflection propagating back through the optical system.
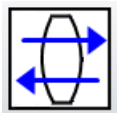
This feature creates a double pass system by reflecting the system about any surface, copying all preceding surface data with pick-up solves as appropriate. The "Reflect At Surface" becomes a mirror. All surface data beyond this surface is lost, all data preceding this surface is reflected and copied with pick-up solves to create a double pass system.
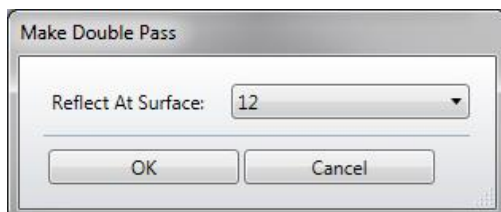
Settings:

Reflect At Surface The surface at which to insert a mirror and create the reflected system.

Apertures

Put apertures on surfaces.

The following tools are available through the drop-down menu of the Apertures toolbar button.
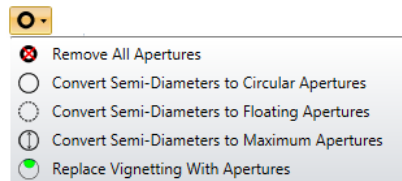
Remove All Apertures

Turns off all surface apertures. All surface aperture types are set to "none" by this tool. Note floating apertures are added automatically to surfaces with power that have user defined semi-diameter values, and so these apertures cannot be deleted; see "Semi-diameters" for more information.

Convert Semi-Diameters to Circular Apertures

Converts all surfaces without surface apertures to fixed semi-diameters with circular apertures set to the semi-diameter value. The primary purpose of this feature is to simplify the analysis of vignetting effects. For most optical designs, it is best to use vignetting factors (see "Vignetting factors") during optimization. However, vignetting factors are an approximation. This feature converts all semi-diameters to surface apertures. The vignetting factors may then be deleted (this is not done automatically by this feature) and the pupil can be overfilled to see where the rays really make it through the system.

Convert Semi-Diameters to Floating Apertures

Converts all surfaces without surface apertures to floating apertures which vignette at the semi-diameter value. This feature is very similar to "Convert Semi-Diameters to Circular Apertures" except floating apertures are used instead of fixed circular apertures. Floating apertures leave the semi-diameter value of the surface in "automatic" mode, and dynamically adjust the vignetting aperture to match the semi-diameter value. Note if any semi-diameters are "fixed" they remain fixed, and all vignetting will occur at the specified semi-diameter on each surface.

Convert Semi-Diameters to Maximum Apertures

Converts all surfaces without surface apertures to semi-diameters set to the maximum required value to pass all rays without vignetting in all configurations. The primary purpose of this feature is to set adequate apertures on all surfaces in zoom lens systems. See "Semi-Diameter: Maximum".

Replace Vignetting With Apertures

Turns off vignetting factors and places surface apertures. This tool uses the identical algorithm as many Zemax analysis features do to replace vignetting factors with surface apertures. Analysis features that need to trace rays from field points in between defined field points need to do this because vignetting is only defined for discrete field points. This tool is useful as a check that Zemax does this conversion properly; and is also handy for removing vignetting factors.

Add Coatings To All Surfaces

Adds any specified coating to all surfaces with air-glass boundaries.

Adds any specified coating to all surfaces with air-glass boundaries. When selected, this tool will prompt for the name of the coating to apply. Any defined coating name may be selected, or choose "none" to remove any existing coatings from surfaces. All surfaces which define a transition from glass to air will have the coating applied (or removed); so this feature is primarily for applying anti-reflection coatings.
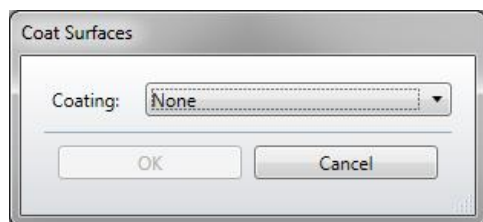
Settings:

Coating Select the coating to be applied to all surfaces with air-glass boundaries.

Go to Surface

Go to the specified surface number
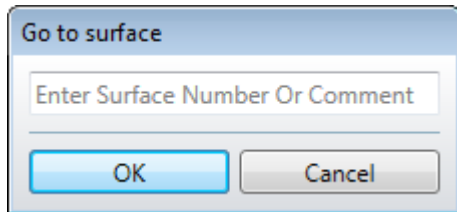
Reset Column Order

Reset column order back to default settings.

Reset Column Widths

Reset column widths to a standard size.

Help

See more help on this feature.

Surface Properties

Clicking the down arrow in the Surface Properties bar above the Lens Data Editor will expand the Surface Properties dialog box. The Surface Properties shown in the dialog box can be changed by clicking on a different surface in the Lens Data Editor, or by clicking the left/right arrows in the Surface Properties bar. The following surface properties may be defined in this dialog box.

For a list of sequential surfaces, see Sequential Surfaces.

Type

Surface type options are set in the Type section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor.

Surface Type

Zemax models planes, spheres, and conics; all of these surface types are grouped under the category of standard surface. On the surface properties dialog box is a list of surface types. Select the appropriate surface type from the drop down list. Zemax supports many different types of surfaces in addition to the standard surface. The types are discussed in Sequential Surfaces. Many optical designs only use the standard surface type.

If the surface type is "User Defined" then the surface shape and ray trace properties are defined in an external program linked into Zemax called a Dynamic Link Library, or DLL. This control selects which DLL the surface uses. See "User Defined" for more information. If the surface type is "Slide" then the bitmap image to apply to the surface is selected with this control.

Surface Color

By default, on the Shaded Model Layout plot, Zemax draws mirrors in green, and refractive and dummy surfaces in blue. The color of the surface as drawn on shaded models may alternatively be selected to be any of the colors defined in the Colors settings, available in the Project Preferences window, which can be displayed via a button in the System section of the Setup tab.

Surface Opacity

If the opacity is set at 100%, then the surface will be rendered on the shaded model plot as a solid color, and the surface may fully obscure other surfaces from view. If the opacity is less than 100%, then the surface is partially transparent, which allows other surfaces to be visible through the partially obscure surface.

Row Color

This control chooses the color of the row in the Lens Data Editor for the surface. By default, glass surfaces, coordinate breaks, mirrors, and paraxial lenses are color coded. Any surface may use either no color, the default color, or a user defined color. The user defined colors are described in "Colors". The coloring of rows may be disabled, see the "Editors" section of the Project Preferences. The entire spreadsheet may be set to show default or no colors, see "Menu options".

Make Surface Stop

The system aperture is the overall system F/#, Entrance Pupil Diameter, Numerical Aperture, or Stop Size. Any of these 4 quantities is sufficient to define the other 3 for a particular optical system. The system aperture is used to define the object space entrance pupil diameter, which in turn is used to launch all rays. The system aperture is always circular. Rays may be vignetted after being launched by various surface apertures. There is only one system aperture, although there may be many surface apertures.

The stop surface may be any surface in the system, except for the object and image surfaces. To change the stop surface, select the "Type" tab and then click on the checkbox labeled "Make surface stop" and press OK. The dialog box will disappear and the surface now will display the "STO" label instead of the surface number. This control is grayed out if the surface is the object, image, or is already the stop surface.

It is important to define the stop surface such that the entrance pupil is on the same axis as the object surface. You can ensure this condition by placing the stop surface of the optical system before any coordinate breaks, obscuration decenters, holograms, gratings, or other components which can alter the optical axis. If your system is symmetric for rotations about the optical axis, then this limitation does not apply. Only systems that use surfaces which can tilt or decenter the optical axis should have the stop placed before any such surfaces. If coordinate breaks are used, but only to implement fold mirrors in an otherwise axial system, then the pupil locations will be correctly computed even if the stop is placed after the fold mirrors.

In certain systems it is not possible to place the aperture stop before coordinate breaks. In this case, ray aiming must be used. Ray aiming is discussed in "Ray Aiming".

Make Surface Global Coordinate Reference

The global coordinate reference surface may be specified in the Type section of the Surface Properties window, or in the Miscellaneous section of the System Explorer.

Global coordinates are defined by a rotation and translation from the local coordinates on each surface. The rotation matrix and the offset vector can be computed for any surface using any other surface as a global reference. The default reference surface is 1, but any other surface may be chosen. The only exception occurs when the object is at infinity, and then 0 may not be the reference surface. The surface selected determines the global coordinate system origin location and orientation. The reference surface is also used to define the point of overlap for multiple zoom positions on the 3D layout plot. See "Global Coordinate Reference Surface" for more information.

Surface Cannot Be Hyperhemispheric

Zemax normally detects if a surface must be hyperhemispheric (the surface continues past the maximum possible radial aperture and curves back toward the vertex to fill more than a hemisphere) to pass all rays. If this option is checked, the surface is not allowed to become hyperhemispheric. This switch should be used in conjunction with a floating or circular aperture to vignette any rays which intercept beyond the desired aperture.

Ignore This Surface

If checked, the surface will be ignored for most purposes. Ray tracing, analysis, layout plots, optimization, tolerancing and most other features will produce results as if the surface had been deleted. The purpose of this feature is to completely ignore surfaces, or groups of surfaces, particularly in multi-configuration lenses where optics are present in some configurations but not others. The multi-configuration operand IGNR can be used to change the ignore surface setting for each configuration.

When a surface is ignored, all properties of the surface are ignored; including radius, thickness, glass, apertures, surface tilts and decenters, coordinate break data, and refraction. However, the surface is not actually deleted, and therefore subsequent surfaces are not renumbered. The ignored surfaces retain their specific surface numbers. Some features which are or may be surface number specific, such as spot diagrams, can still be computed on the ignored surfaces. The resulting data will be for the first surface that is not ignored following the requested surface. Some features that list text data on a surface by surface basis may still list data for ignored surfaces, but the data on subsequent surfaces that are not ignored will be as if the ignored surfaces had been deleted.

Draw

The settings for how surfaces are drawn are selected in the Draw section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor.
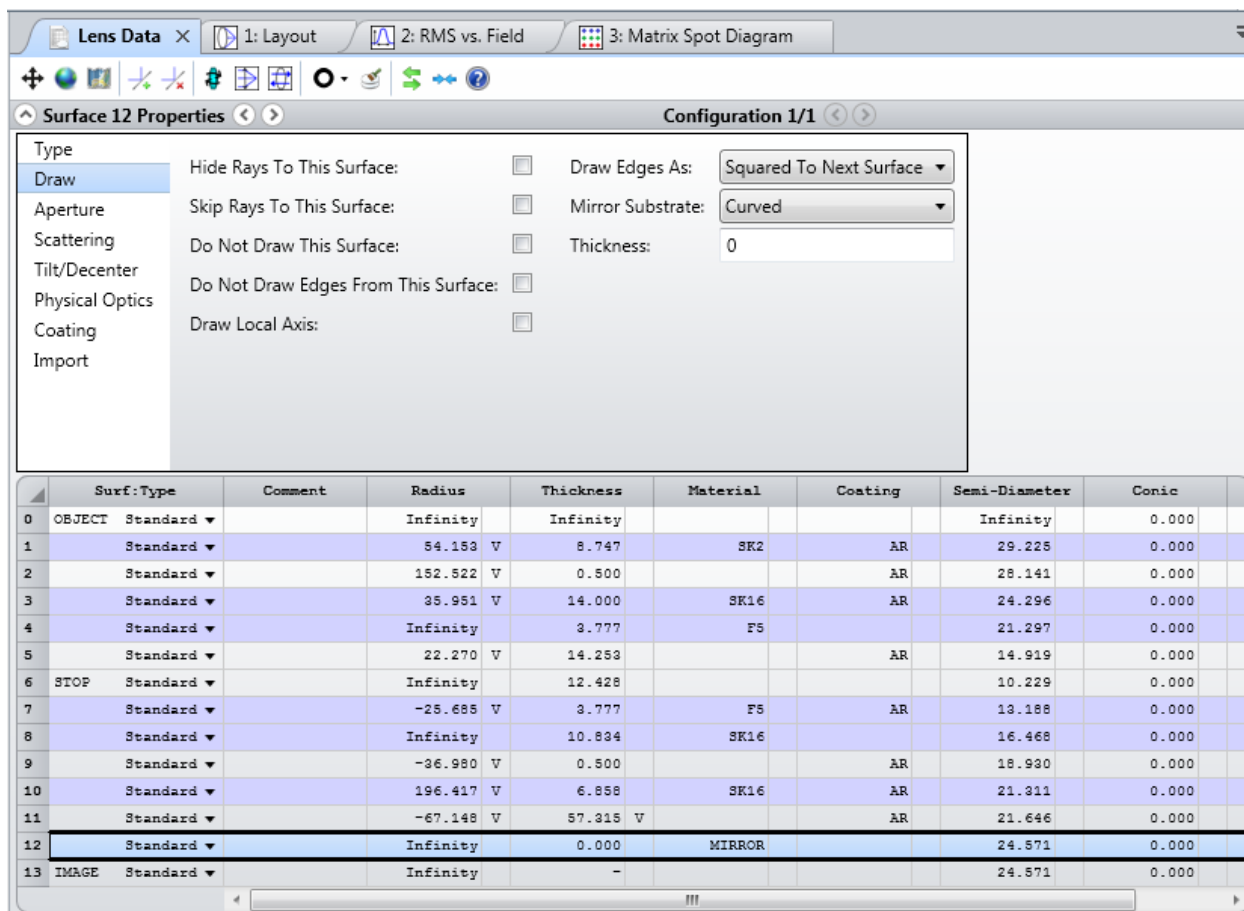
Hide Rays To This Surface

If checked, no rays will be drawn to the surface in the various types of layout plots. See Skip Rays below for an alternate means of hiding rays to dummy surfaces.

Skip Rays To This Surface

If checked, rays will skip the surface. This means ray segments will be drawn from the last surface that was not a coordinate break and did not have the Skip Rays option checked, all the way to the next surface that is not a coordinate break and does not have the Skip Rays option checked. For example, if Skip Rays is selected on surfaces 3, 4, and 5, rays will be drawn directly from the coordinates on surface 2 to their coordinates on surface 6, as long as surfaces 2 and 6 are not coordinate breaks.

The purpose of the Skip Rays option is to suppress the drawing of rays to and from dummy surfaces. Although this can be useful in some cases, this option can cause strange, misleading layout plots if the skipped surfaces can refract or diffract the rays. Therefore, the option should only be used on dummy (optically ineffective) surfaces. Zemax does not check or verify that surfaces with the Skip Rays option are in fact dummy surfaces, so care is required when using this feature.

Do Not Draw This Surface

If checked, the surface will not be drawn on layout plots.

Do Not Draw Edges From This Surface

If this option is checked, then no edges will be drawn from this surface to the next surface. This feature makes for cleaner drawings in some systems, especially those using liquids or other non-air fillers between optical surfaces.

Draw Local Axis

If checked, an arrow will be drawn to indicate the orientation of the local Z axis at the surface on 3D Layout plots. The size of the arrow is determined by the fletch size, see "Fletch Size".

Draw Edges As

This setting controls how edges that connect a surface to the next surface are drawn on the various layout and element drawings. Usually this setting is only used to connect edges on surfaces with circular apertures defined by the radial semi-diameter. Other cases may use another, usually simpler algorithm than the options described here, or may not draw the edge at all. The available options are described below:

Squared To Next Surface A flat face is drawn radially outward from the smaller of the current and next surfaces to match the semi-diameter of the larger surface. The outer edge of the surface is then formed by a cylinder extending to connect the surface edges.

Tapered To Next Surface The current and next surface are connected by a single tapered cylinder.

Flat To Next Surface The current surface edge is drawn as a cylinder at the current semi-diameter value extending to the contact point on the next surface.

The 3D Layout, Wireframe, Shaded Model, and Solid Model plots currently do not support the "Flat To Next Surface" option. Surfaces with this setting selected will be drawn using the "Squared To Next Surface" method.

Mirror Substrate and Thickness

Sequential surfaces with a glass type of "MIRROR" may be drawn as either a thin surface, or as a solid object with a substrate. The back of the substrate may be a flat surface that is perpendicular to the local z axis, or as a curved surface that follows the contour of the actual mirror surface. The thickness of the substrate is measured along the local z axis at the vertex of the surface, even if the surface aperture is decentered or user defined. Zemax will draw the edges of the substrate by connecting the mirror surface and the substrate back. All surface and aperture types are supported on all layout plots.

By default, Zemax will draw the back of the mirror substrate as a curved surface with a thickness of 4% of the surface semi-diameter, or 0.5 lens units if the semi-diameter is zero. The default thickness will be used if the Mirror Substrate Thickness parameter is set to zero.

Mirror substrates will not be drawn on surfaces which are mirrors in some, but not in all defined configurations. Likewise, they will not be drawn on mirrors that are only set to draw in specific configurations. Mirror substrates are also not drawn on surfaces which are Mangin mirrors (mirrors on the back of glass substrates) in any configuration. To draw mirror substrates on surfaces which are present in some configurations but not in others, use the multi-configuration operand IGNR to ignore the mirror surface rather than GLSS to change the surface from a mirror to glass or air. See "Adding and removing elements".

Aperture

Setting aperture data for each surface is done in the Aperture section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor
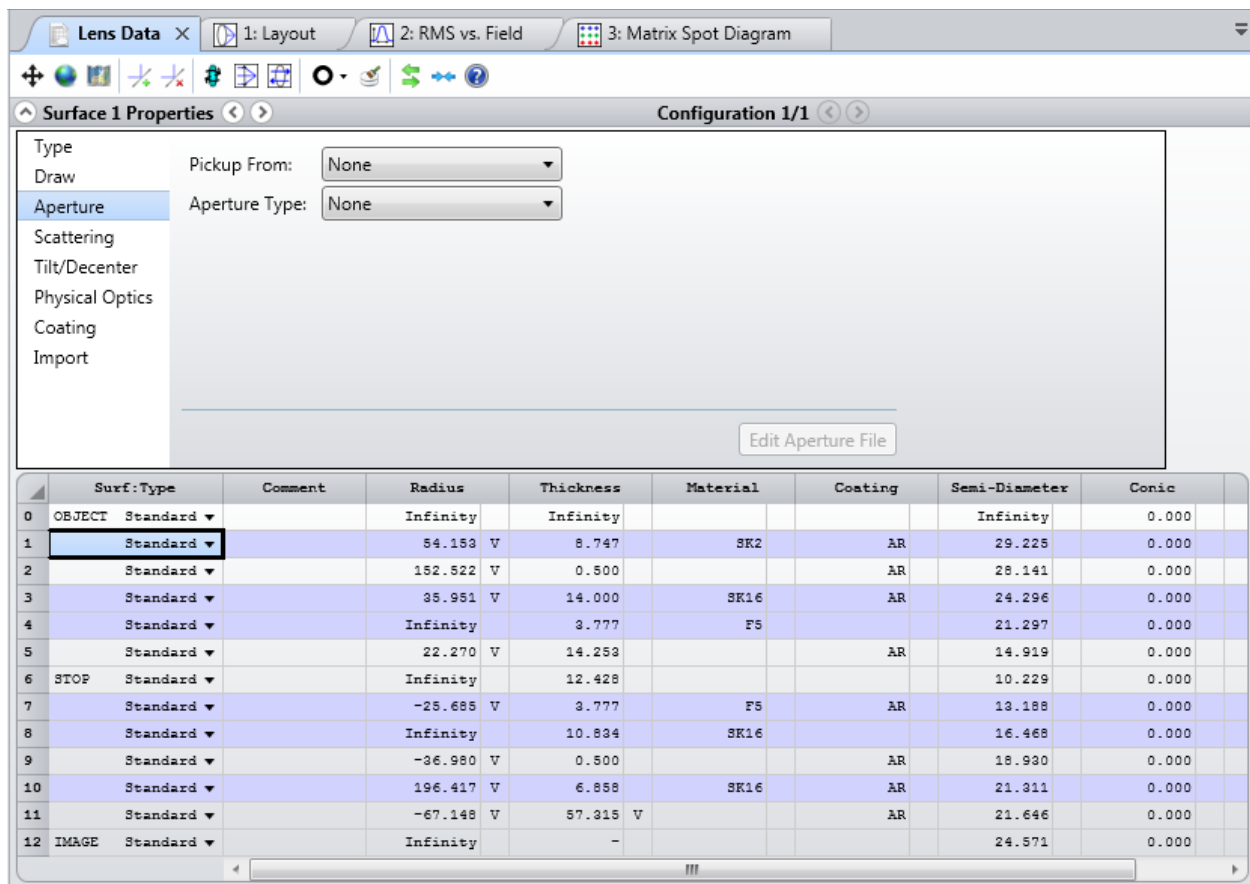
Surface apertures are used to account for the effects of vignetting. The aperture and obscuration types define regions which pass or stop rays, respectively. More than one aperture may be described at a given optical element by inserting a dummy surface with zero thickness at the desired location, and then setting additional apertures on that surface. This is useful for constructing complex apertures. Multiple simultaneous apertures and obscurations may also be defined on a single surface using the user defined apertures and obscurations feature, see "User defined apertures and obscurations".

Aperture decenters and pickups

All types of apertures may be decentered from the current optical axis by specifying either X- or Y-decenters, or both. The decenters are given in lens units. If the identical aperture is used on more than one surface, the aperture "Pickup From" is a useful feature. Any aperture may be picked up from any previous surface.

Aperture Type

None When "None" is selected for the aperture type (the default), all rays which can be refracted or reflected by this surface are allowed to proceed. To clear any aperture to this default state, or to change the current aperture type, select another aperture type.

Circular Aperture/Obscuration A Circular Aperture defines an annular region which vignettes all rays which strike the surface inside of the minimum radius, and outside of the maximum radius. If the ray is between the minimum and maximum radii, then the ray will be allowed to proceed. The Circular Obscuration is the complement of the Circular Aperture.
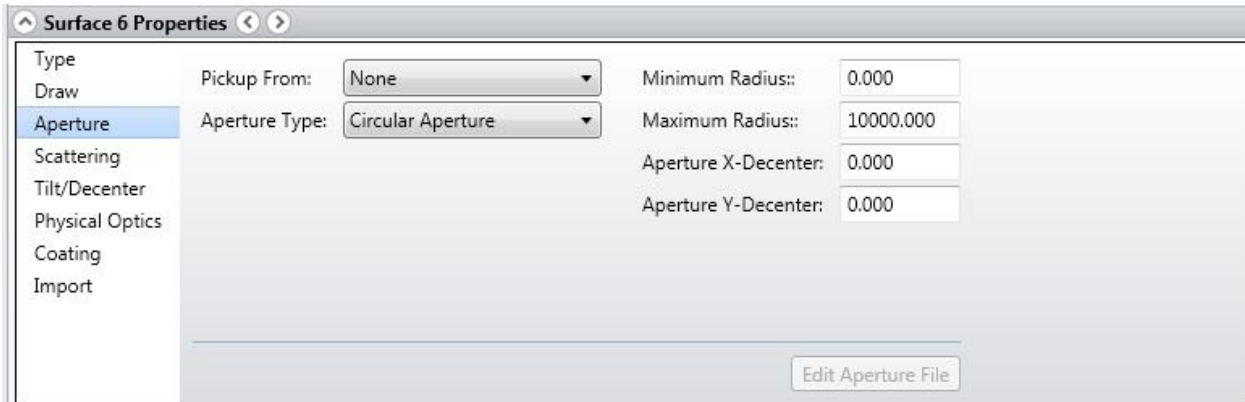
Rectangular Aperture/Obscuration Rays are vignetted which intercept the surface outside the rectangular region defined by the half widths in x and y. The Rectangular Obscuration is the complement to the Rectangular Aperture.
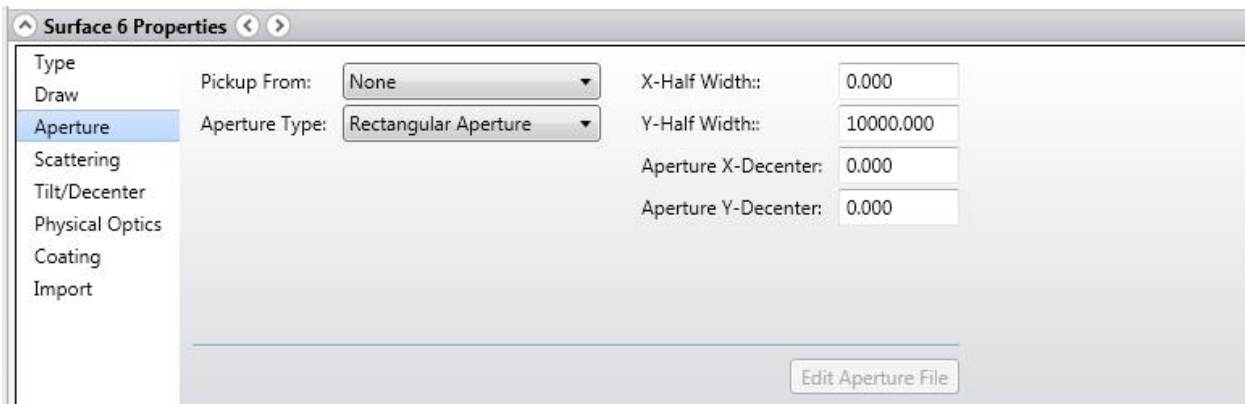
Elliptical Aperture/Obscuration Rays are vignetted which intercept the surface outside the elliptical region defined by the half widths in x and y. The Elliptical Obscuration is the complement to the Elliptical Aperture.
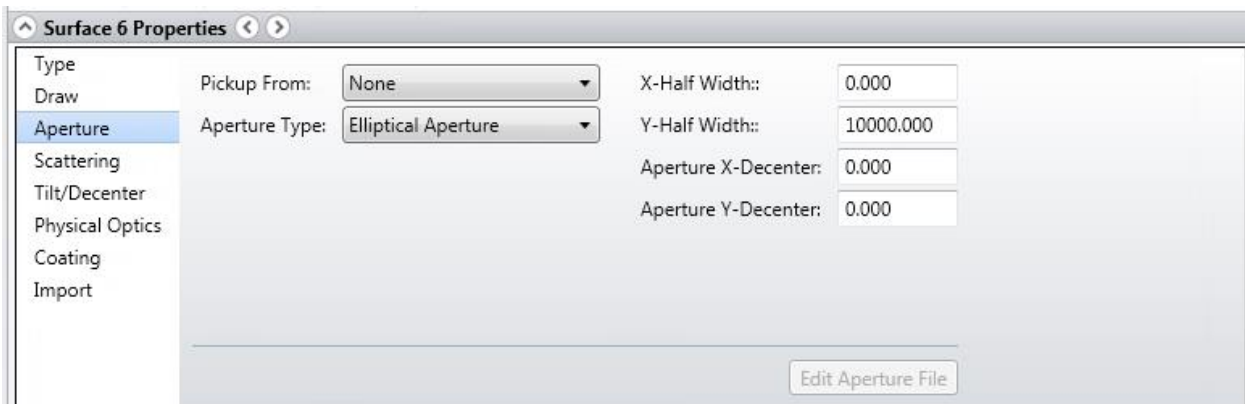
Spider The spider is defined by the width of each arm, and by the number of arms. Zemax assumes the arms are all the same width, and that they are spaced in equal radial angles. The first arm starts at a radial position of zero degrees, which is along the local positive x-axis. More complex spiders, which contain arms of different widths at unequal angles can be constructed using several spiders on adjacent dummy surfaces. The coordinate break surfaces can be used to rotate the spider(s) to any desired angle.
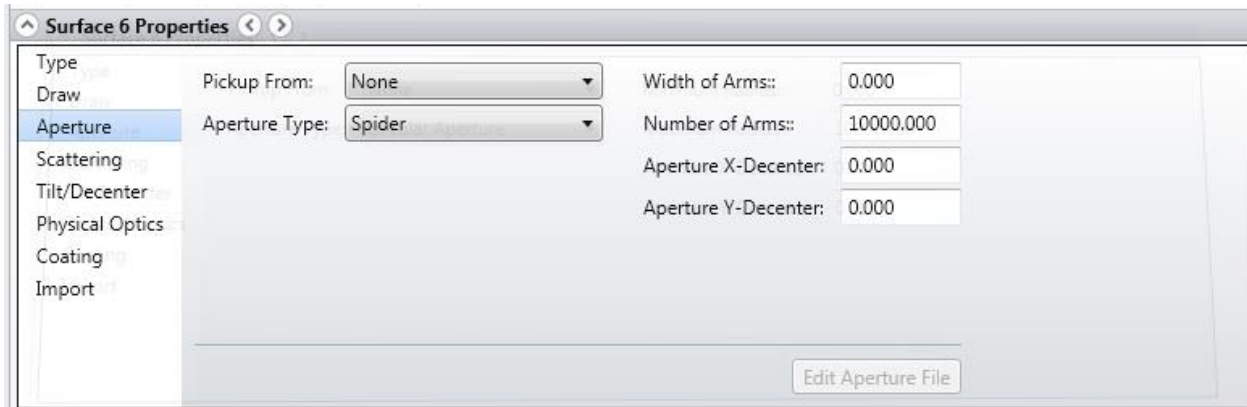
User Defined Aperture/Obscuration Zemax supports a general user specified aperture defined by a series of line segments, arcs, circles, polygons, and rectangles. The aperture may be closed in a simple or complex way, and multiple aperture regions may be defined which are either nested or not nested. To define a user defined aperture or obscuration, select the desired type (aperture or obscuration) from the list of aperture types. The "Aperture File" control will list all available user defined aperture (UDA) data files.

The UDA files are text files that may be created and edited using any text editor. The UDA files are stored in the <objects>\Apertures folder (see "Folders"). The button "Edit Aperture File" will invoke a text editor to allow user editing of the selected UDA file. The UDA file needs to saved and the lens file updated to make the changes effective. The UDA Scale is a dimensionless multiplier that scales the aperture defined in the UDA file. This control allows scaling of the UDA without need to modify the UDA file. See "User defined apertures and obscurations".

Floating Aperture A floating aperture is very similar to the circular aperture, except the minimum radius is always zero, and the maximum radius is always equal to the semi-diameter of the surface. Since the semi-diameter value may be adjusted by Zemax (when in automatic mode) the aperture value "floats" as the semi-diameter value. The floating aperture is useful when macros or external programs use Zemax to trace rays that may lie outside of the default semi-diameters, and these rays are to be vignetted.

Aperture projection on non-plane surfaces

All apertures are modeled as a surface projected from the vertex tangent plane onto the optical surface. The actual ray-surface x and y intercept coordinates are used to determine vignetting, the z coordinate is ignored. Different results may be calculated for steep optical surfaces if the aperture is placed on a dummy surface in front of the optic, instead of being directly entered on the curved surface. This will only occur if the ray incidence angles are steep. Usually it is best to place the apertures directly on the optical surface, unless of course the dummy surface more accurately represents the situation.

User defined apertures and obscurations

The circular, rectangular, and elliptical apertures and obscurations are easy to use and cover common cases. However, there are times when a more general aperture model is required.

Zemax supports a general user specified aperture defined by a series of line segments, arcs, circles, polygons, and rectangles. The aperture may be closed in a simple or complex way, and multiple aperture regions may be defined which are either nested or not nested. To define a user defined aperture or obscuration, select the desired type (aperture or obscuration) from the list of aperture types. The "Aperture File" control will list all available user defined aperture (UDA) data files. The UDA files are text files that may be created and edited using any text editor. The UDA files are stored in the <objects>\Apertures folder (see "Folders"). The UDA file format is described in the next section.

The button "Edit Aperture File" will invoke a text editor to allow user editing of the selected UDA file. The UDA file needs to saved and the lens file updated to make the changes effective. The UDA Scale is a dimensionless multiplier that scales the aperture defined in the UDA file. This control allows scaling of the UDA without need to modify the UDA file.

The UDA file format

The UDA file consists of a series of simple text commands that define pieces of the boundary shape of the aperture. The individual pieces are called entities. All UDA entities are defined as they lie on the XY plane. The Z coordinate is determined by the sag equation of the surface to which the UDA is applied. The commands which define the entities consists of three letter mnemonics followed by numerical values stored in a text file whose extension ends in UDA. The file name may not include any spaces.

In the entity definitions, all spatial coordinates are in lens units and all angles are in degrees. The available entity types are described in the following table.

USER DEFINED APERTURE (UDA) ENTITIES

Entity Name/Syntax	Description
ARC cx cy angle n	Arc. Appends an arc to the current end point. The starting point of the arc is the current end point, defined by a previous ARC or LIN command. The cx and cy values are the center of the circle on which the arc lies. Note that the current end point, along with cx and cy, define the radius of the arc. The angle (in degrees) defines the length of the arc. Positive angles are clockwise on the XY plane. Zemax internally represents the arc as a series of line segments. The parameter n determines how many line segments are used to approximate the arc. A recommended value is roughly 1 segment for each 6 degrees of arc. The maximum allowed value is 64 segments.
BRK	Break. Ends the current aperture definition, closing any previously defined aperture. Entering the coordinates: 0.0 0.0 or LIN 0.0 0.0 will also be interpreted as a break.
CIR cx cy radius n	Circle. Defines a circle centered on cx and cy, with the specified radius. Zemax draws the circle as a series of n line segments. The minimum value of n is 8. The maximum value of n is 64. The default value is 32 if n is omitted. The actual aperture applied to

	the rays is an exact circle, independent of the value for n.
ELI cx cy rx ry angle n	Ellipse. Defines an ellipse centered on cx and cy, with the specified radial (half) widths in the x and y directions. Zemax draws the ellipse as a series of n line segments. The minimum value of n is 8. The maximum value of n is 64. The default value is 32 if n is omitted. The actual aperture applied to the rays is an exact ellipse, independent of the value for n. The parameter angle (in degrees) may be used to rotate the ellipse clockwise to any desired position.
LIN x y n or x y	Line. Appends a line segment from the current end point to the specified x, y coordinates. Use BRK to close a polygon formed by LIN commands. Zemax optionally represents the line as a series of shorter line segments; this is useful when the line lies on a steeply contoured part. The parameter n determines how many line segments are used to construct the total line. If n is omitted then 1 line segment is used. The mnemonic LIN is not required; if two coordinates are listed with no mnemonic, LIN is assumed. This is for backward compatibility with older versions of Zemax.
POL cx cy radius n angle	Polygon. Defines an "n" sided regular polygon centered on cx and cy, with each vertex the specified radial distance from the center point. The POL command may be used to define triangles, squares, pentagons, hexagons, and other polygons with equal length sides. The first point on the polygon starts at the angle 0.0 (along the local +x axis) and proceeds clockwise from there. The parameter angle (in degrees) may be used to rotate the polygon to any desired position. The maximum allowed number of segments is 64, the minimum is 3.
REC cx cy xhw yhw angle nx ny	Rectangle. Defines a rectangle centered on cx and cy with the specified half widths in the x and y directions. The parameter angle (in degrees) may be used to rotate the rectangle clockwise to any desired position. The optional arguments nx and ny define how many segments to use in the x and y directions. Using a larger number of segments will yield better rendering of NSC object surfaces that are steeply curved over the rectangle aperture.
! comment	Comment. Any line starting with the ! symbol is ignored, and may be used for placing comments in the UDA.

The end of a polygon is indicated by either a BRK entity or a LIN entity with both X and Y being set to zero. For this reason, a polygon may not be defined which has a vertex at (0,0), or within a tolerance distance on the order of 1.0E-06 to that origin coordinate. If a vertex must be defined at (0,0), a work around is to instead use some very small (but larger than the tolerance) value for one of the coordinates, such as (1.0E-05, 0) As long as at least one of the coordinates is not zero, the point will be considered a vertex rather than indicating the end of a polygon. The last listed vertex in the

polygon is assumed to then connect to the first point. Note that because the last vertex is automatically connected to the first vertex, the last vertex should not be defined to be identical to, and redundant with, the first point. Defining multiple points that are within round-off error of the same location may result in incorrect ray tracing or aperture rendering.

Note that the UDA entities REC and CIR define "stand alone" apertures not connected to any other aperture entities, while LIN and ARC define pieces of a polygon whose definition ends with a BRK. When using LIN and ARC entities, the first entity defined should be an LIN, which defines the first point of the polygon. Multiple apertures may be defined separated by BRK entities.

The maximum number of line segments defined within the UDA file is limited only by available memory. However, ray tracing and rendering of surfaces and objects is slower when using very complex UDA data files. See the following sections for UDA examples.

UDA Examples

To define a square 20 lens units on the side, the UDA file is:

```
LIN -10, -10
LIN -10, 10
LIN 10, 10
LIN 10, -10
BRK
```
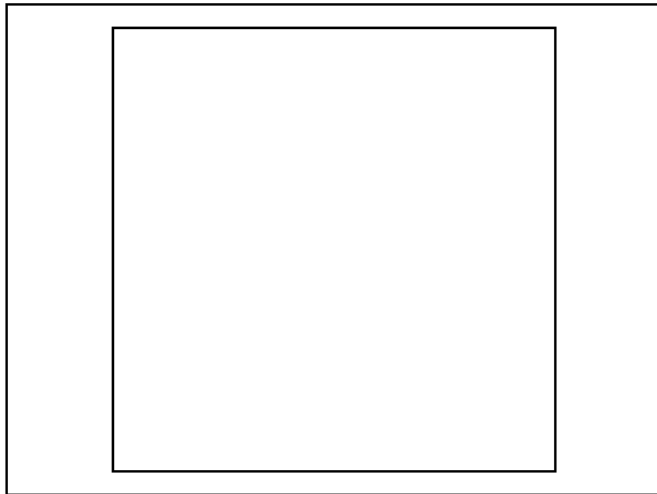

Note the BRK indicates the end of the polygon definition, and the last point is assumed to join back to the first point, thus defining the last side of the square. This same aperture may be defined with a single REC command:

```
REC 0 0 10 10 0
```

Note the REC command does not require a BRK command to define the end of the aperture.

Multiple polygons may be defined by separating them with a single BRK. For example, to define an aperture consisting of two slits, each 5 lens units wide, with an inner separation of 10 lens units, the UDA file is:

```
LIN -10, -10
LIN -10, 10
LIN -5, 10
LIN -5, -10
BRK
LIN 10, -10
```

```
LIN 10, 10
LIN  5, 10
LIN  5, -10
BRK
```

Or the equivalent

```
REC -7.5 0 2.5 10 0
REC +7.5 0 2.5 10 0
```
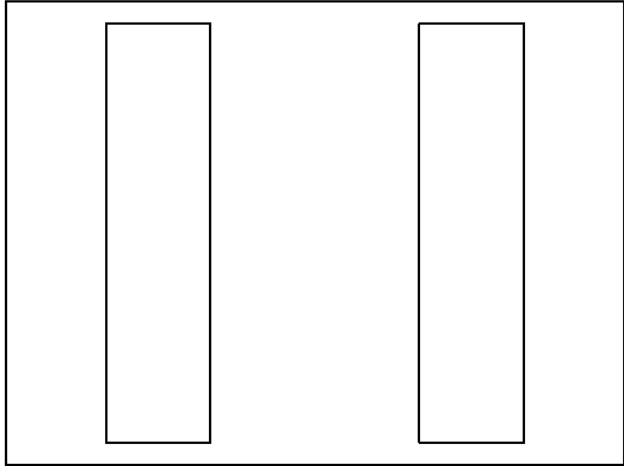

To create a square with rounded edges, use the ARC and LIN commands together. Here is a UDA that creates a square, 4 lens units on a side, with rounded corners. Each rounded corner is formed by one ARC with a radius of 1 lens unit and 12 segments along the arc:

```
LIN -1 2
LIN 1 2
ARC 1 1 90 12
LIN 2 -1
ARC 1 -1 90 12
LIN -1 -2
ARC -1 -1 90 12
LIN -2 1
ARC -1 1 90 12
```
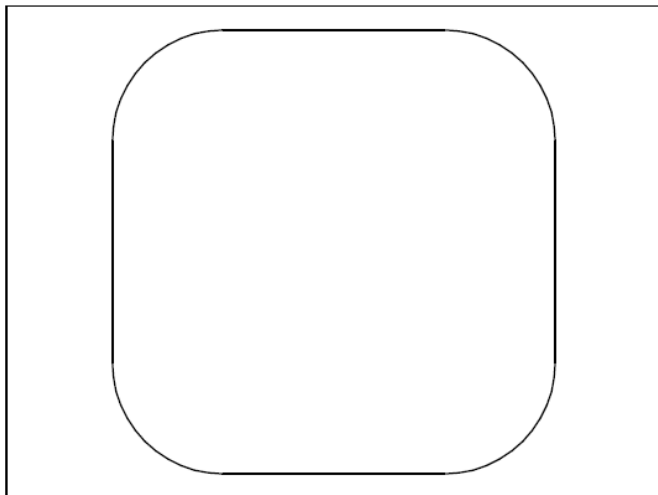
The first LIN command defines the starting point near the upper left corner. The subsequent commands define each of the 8 parts of the aperture. Note a BRK command is not needed if only a single aperture region is defined.

Multiple polygons may also be nested. If a ray intercepts a point within a polygon that is within another polygon, then the point is considered to be outside the aperture. This convention allows "islands" to be defined within apertures that become obscurations, and vice-a-versa. Any number of nesting levels is allowed, and each level toggles the inside/outside status of the point.

Note that when multiple polygons are defined, the polygons may be separated or nested. However, they may not intersect or share adjacent boundaries.

Here is a UDA with the rounded corner square from the previous example, with 5 small sub apertures nested within; a pentagon, a hexagon, an ellipse, a rectangle, and a circle. Note the BRK command is not needed to separate the REC, ELI, POL, and CIR entities from the ARC and LIN entities because REC, ELI, POL, and CIR all define stand alone aperture regions:

```
LIN -1 2
LIN 1 2
ARC 1 1 90 12
LIN 2 -1
ARC 1 -1 90 12
LIN -1 -2
ARC -1 -1 90 12
LIN -2 1
ARC -1 1 90 12
REC 1 1 .3 .5 -30
ELI -1 -1 .6 .2 15
POL 1 -1 .5 5 0
POL -1 1 .5 6 0
CIR 0 0 .3
```

Scattering

Scattering options are set in the Scattering section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor.
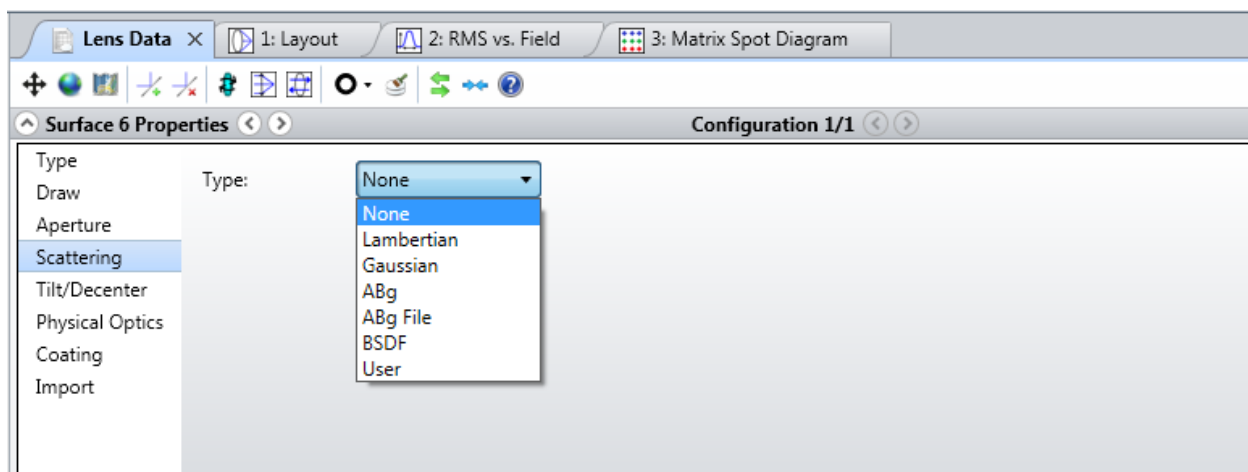
Highly polished optical surfaces typically have a small amount of scattered light over a small, roughly angular cone around the specular refracted or reflected ray direction. Surface scattering properties are useful for modeling this small

amount of scattered light, to analyze the effects of the scattering on MTF or other quality measures. Scattering is modeled in Zemax by randomly deviating the refraction or reflection angle of some or all of the rays leaving a surface.

There are two fundamentally different ways of modeling scattering in Zemax: sequential and non-sequential.

For the sequential model, the scattering is assumed to deviate the ray angle only very slightly. The primary effect is a "blurring" of the spot diagram or ray pattern at the image surface. Sequential scattering is not intended for modeling the effects of back scatter, wide angle scatter (such as Lambertian) or other scatter where the scattered rays may not follow a well behaved sequential path to the image surface.

For the non-sequential model, scattering may occur over any range of angles, and rays will be traced correctly no matter in what direction they propagate. The non-sequential model is the better choice if it is necessary to model scatter from surfaces that are not part of the sequential ray path, such as from lens mounts, baffles, or other objects. See "NSC Ray Tracing" under the "Non-Sequential Components Editor" section of the Setup Tab help file for details on this method of modeling scatter. It is not necessary to use the non-sequential components feature of Zemax to model scattering, if the system is otherwise sequential and the only scattering of interest is small angle scatter from the small scale roughness of the optical surfaces, and the only desired analysis is the degradation in image quality due to scattering.

Scattering Types

Scatter models are defined in terms of a probability distribution function. When Zemax scatters a ray, a new direction of propagation is chosen. The direction is chosen using a probability function and one or more random numbers. The net effect is that if many rays are traced, the resulting scattered ray distribution would approach the probability distribution function. There are seven scattering models available: none, Lambertian, Gaussian, ABg, ABg File, BSDF, and user defined. Each available scatter model is summarized below, but see "Scattering" for the detailed technical descriptions.

None The default scatter model is "None", which means no scattering will occur. The resulting ray is called the unscattered or the "specular" ray (even if the surface is not actually reflective). The Bi-Directional Scatter Distribution Function (BSDF), which is the scattered radiance per unit incident irradiance, is zero.

Lambertian In the Lambertian scattering model, the scattered ray projection vector has equal probability anywhere in the unit circle, and the BSDF is just $1/\pi$. The scattered intensity is proportional to the cosine of the angle between the normal vector and the scattered ray angle. Note Lambertian scattering is independent of the ray incident angle. Most diffuse surfaces are nearly Lambertian. Although Lambertian is a valid option on the sequential surface dialog, rays which scatter using this model may go in any forward direction, which may cause rays to scatter at large enough angles so that they do not correctly propagate through the rest of the optical system.

Gaussian In the Gaussian scattering model, the scattering distribution is rotationally symmetric in direction cosine space, no matter what angle the specular ray makes with respect to the surface normal. The BSDF expression (see "Scattering") contains a dimensionless value σ, which determines the width of the Gaussian distribution on the projected plane. Values of σ greater than about 5.0 yield a BSDF that is nearly Lambertian. For this reason, the maximum allowed value of σ is 5.0.

ABg Model The ABg scattering model is a widely used method for defining the BSDF. This scattering model is generally a good model to use when the scattering is mainly due to random isotropic surface roughness, and the scale of the roughness is small compared to wavelength of light being scattered. These assumptions are generally valid for polished optical surfaces. See "Scattering" for a detailed technical description.

ABg File The ABg File scattering model allows a sum of ABg profiles to be used to define the scattering properties of a surface. The profiles to use are specified in a text file. The text file must have a .ABGF extension and must be located in the <data>\ABg_Data folder (see "Folders"). All of the profiles specified in the ABGF file must be defined in the currently loaded ABg Data File (see "ABg Data File"), and all profiles must be specified using upper case letters (regardless of capitalization of the profile name in the ABg Data File).

BSDF The BSDF scattering model allows the use of tabular BSDF data for defining the scattering properties of a surface. Data are provided via text files. Files must follow the BSDF Data Interchange file format described in the article

entitled "BSDF Data Interchange File Format Specification", which may be found in the Zemax Knowledge Base. Zemax allows definition of separate BSDF data for reflection and refraction. If the specular ray reflects or refracts, the ray is subsequently scattered using data from the appropriate input file. For a file to be available for use in this scatter model, it must have a .BSDF extension (as indicated in the article describing the file format), and be located in the <data>\Scatterdata folder (see "Folders"). A full description of the model and its use is provided in the Knowledge Base article entitled "How to Use Tabular Data to Define the Surface Scattering Distribution" also available on the Radiant Zemax web site.

User defined scattering Completely general surface scattering may be defined via an external program called a Dynamic Link Library (DLL). Sample DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs" and "Scattering – User defined scattering" for a detailed description.

The non-sequential ABg, ABg File, and BSDF scattering models support both reflection and refraction coefficients, while the equivalent sequential scattering models use a single set of coefficients since sequential surfaces either reflect or refract, but not both. Whether a sequential surface reflects or refracts depends only upon the whether the surface is a mirror or not.

Many sequential analysis features have a "Use Polarization" option to consider ray transmission. When using polarization, all sequential surfaces that scatter rays should be defined to have small-angle scatter. Small-angle scatter means the rays are only perturbed in direction by an amount that would not significantly change the polarization properties of the ray. The reason for this requirement is that the polarization transmission computation is performed only on the unscattered ray. The unscattered polarization transmission is then applied to the actual scattered ray. This limitation only applies to sequential systems. For systems that require large angle scatter and polarization analysis, use the non-sequential component capability as described in "NSC - OVERVIEW".

Only some Zemax analysis features use the scattering data. Most features, such as optimization and layout diagrams, ignore surface scattering. Features that use the scattering data, such as the spot diagrams have a "Scatter Rays" checkbox.

Scatter Fraction

If a scattering model other than "None", "ABg", or "ABg File" is selected, the "Scatter Fraction" must be defined. This fraction must be between 0.0 (no rays will be scattered) and 1.0 (every ray will be scattered). For the ABg and ABg File scattering models, the fraction to scatter is determined by the ABg parameters, see "Defining ABg data".

The decision to scatter or not to scatter is made by the generation of a single random number between 0.0 and 1.0. If this random number is larger than the fraction to scatter, the ray will not scatter, otherwise, the ray will scatter. For example, if the fraction to scatter is 1.0, the ray always scatters. If the fraction to scatter is 0.0, the ray will never scatter. If the

fraction to scatter is 0.25, then on average one out of four rays will scatter. All of the energy of the ray follows the randomly generated scatter path.

Tilt/Decenter

Surface tilt and decenter options are set in the Tilt/Decenter section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor.
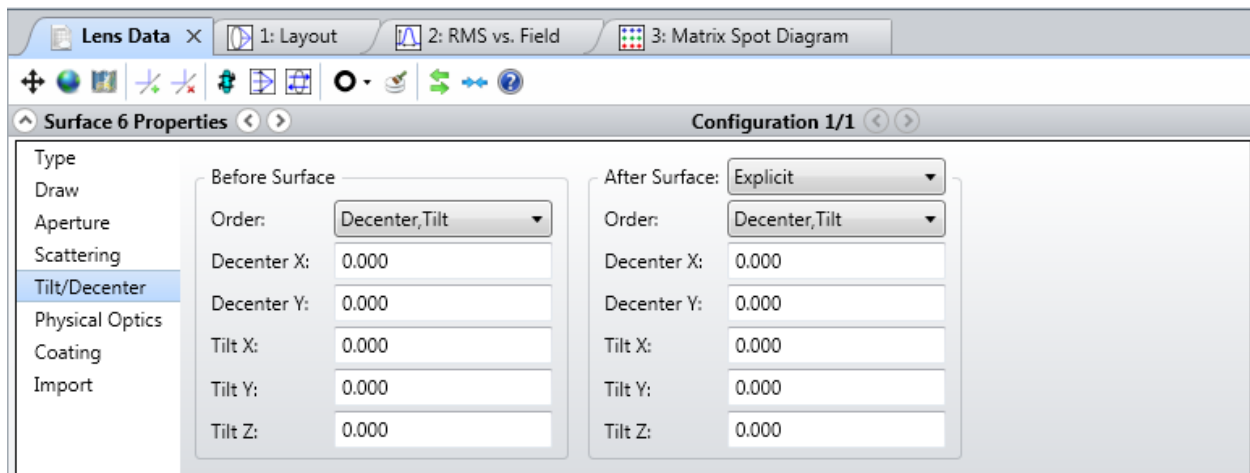
Surface tilts and decenters allow a change in the coordinate system to be implemented both before and after ray tracing to the surface. Applications include decentering a surface and returning to the original coordinate system, tilting a fold mirror and tilting again to follow the beam, tilting a surface to model a wedge, and many others. Surface tilts and decenters are redundant with and very similar to coordinate breaks. For detailed information on coordinate breaks, see "Coordinate Break". A surface tilt/decenter can be thought of as a coordinate break, followed by the surface, followed by another coordinate break.

Surface tilts and decenters may not be placed on a coordinate break surface. However, coordinate break surfaces support a coordinate return solve, see "Using the Coordinate Return". Surface tilts and decenters are also not allowed before the first surface if the object is at infinite conjugates.

The advantage to using surface tilts and decenters is the elimination of "dummy" coordinate break surfaces in the Lens Data Editor. This allows for a somewhat less cluttered display some users prefer. The disadvantage of using the surface tilts and decenters is the current implementation does not support optimization of surface tilt and decenter data.

The operation of surface tilts and decenters is done in a sequence:

Before the surface, the coordinate system is decentered in X, decentered in Y, and then tilted around X, tilted around Y, tilted around Z. Decenters are measured in lens units, tilts in degrees about the respective axis in a right hand direction. Alternatively, the order may be chosen to tilt then decenter. In this case, the coordinate system is tilted around Z, tilted around Y, tilted around X, decentered in Y then finally decentered in X.

After the surface, the same set of operations may be performed in either order. The values for the before and after decenters and tilts, and the order in which they are done, may be independent. However, it is frequently useful to have the after tilts and decenter values be related to previous before values. The options available are:

a) Explicitly define the tilts and decenters.

b) Pickup the values from the before data of the current surface (typically used for fold mirrors)

c) Reverse the values from the before data of the current surface (sometimes called a decenter/ tilt and return)

d) Pickup the values from the before data of a prior surface

e) Reverse the values from the before data of a prior surface (typically used for decentering a range of surfaces)

All of these options are supported by the "After Surface" settings. Reversing values from a prior surface involves changing the order and picking up the tilt and decenter values from the target surface and reversing the sign.

The coordinate system resulting from the before and after tilt and decenters will define the coordinate system for the next surface. The thickness of a surface is the thickness in the new coordinate system after all the tilt and decenters are applied, measured along the resulting Z axis.

Physical Optics

The Physical Optics options are set in the Physical Optics section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor.
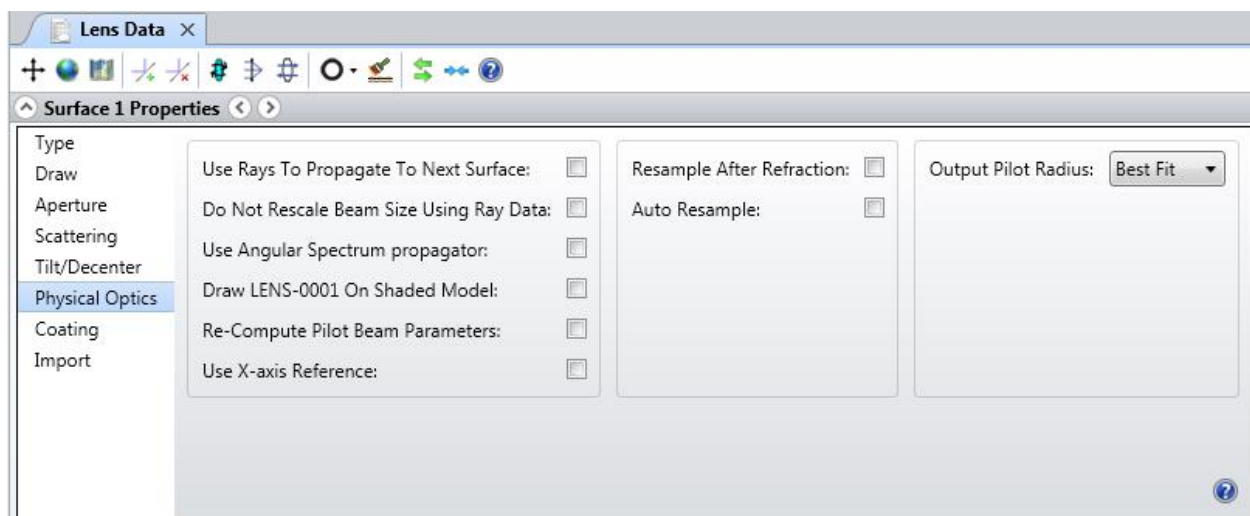
Use Rays To Propagate To Next Surface

If checked, then the diffraction propagation algorithms will not be used to propagate the beam to the next surface. Instead, rays will be traced to the next surface and the resulting ray transfer function will be used to propagate the beam and update the pilot beam parameters. Use of this feature is required for non-sequential, birefringent, and gradient index surfaces. Multiple sequential surfaces may all have this option checked, in which case the beam will advance through all the surfaces using only ray propagation. This feature may also be used to speed up the propagation algorithm by propagating through several surfaces at once, if the distance between the surfaces is small enough so that the diffraction propagation is unimportant. The default is unchecked. Search the help files for "Considerations when using rays to propagate" for important information about using this feature near surfaces where the rays come to a focus.

Do Not Rescale Beam Size Using Ray Data

By default, Zemax will use the ray grid to determine any distortion, stretching, scaling or other change in the beam shape. A good example of this is the passing of the beam through a grating. The beam will become compressed along the direction of the diffraction. However, there are times when this computation fails. One such case is when the ray grid

enters a caustic. The rays no longer accurately represent the beam, and should not be used to determine the beam shape. Zemax automatically skips this step if it detects the ray grid is in a caustic, however, there may be other cases where the algorithm should not be applied, and this option allows user selective disabling of the algorithm.

Use Angular Spectrum Propagator

If selected, the angular spectrum propagator will be used for propagation through the surface instead of the propagator Zemax would automatically select. This option should only be used if the beam size does not change dramatically over the propagation distance. The array width will remain constant using the angular spectrum propagator. The default is unchecked.

Draw "lens file name-surface #" on shaded model

If selected, the ZBF of the displayed name will be drawn on shaded model layout plots at the location of the surface. The file name is the same as the name of the lens file, with a 4 digit suffix equal to the surface number. This is the same name convention that automatically saved beam files use. See "Display Tab" for information on saving beam files at every surface. If this option is selected, and the correct beam file name is found, then the surface itself will not be drawn, just the beam file at that surface. For this reason, this option works best on dummy surfaces. Drawing large beam files uses considerable amounts of memory and will slow down the drawing of shaded model displays.

Re-Compute Pilot Beam Parameters

Some surfaces significantly alter the beam characteristics. A good example is a pinhole aperture. After passing the aperture, the beam waist size, divergence, and position may be significantly altered, even though technically the pinhole has no optical power (this is in fact the core difference between geometric and physical optics). After passing the pinhole, the pilot beam parameters need to be recomputed for subsequent accurate propagation. Checking this option will invoke an algorithm which finds the pilot beam parameters that best fit the actual beam.

Use X-axis Reference

By default, the grid of rays used to generate the transfer function at any surface is oriented such that the Y-axis of the grid aligns with the Y-axis of the beam. In highly tilted systems this choice of alignment may not be appropriate. This option allows the orthogonal choice of grid alignment, i.e. one in which the X-axis of the grid aligns with the X-axis of the beam.

Resample After Refraction

This control allows the beam sampling and width to be changed after propagating through any surface. If selected, the new x and y sampling and beam width may be specified. Note that this control is ignored if rays are used to propagate through the surface. See also "Auto Resample" below.

Auto Resample

If checked, the beam will be automatically resampled. The algorithm used first recomputes the pilot beam parameters as described in "Re-Compute Pilot Beam Parameters" above. The X and Y- widths are then set according to the equations described in "Comments about point spacing and sampling". The number of points used in the X- and Y- sampling will not change.

Output Pilot Radius

The output pilot beam is used to reference the phase and other properties of the beam after refraction through the surface. This control defines the method of computation for the phase radius of curvature of the output pilot beam.

Best Fit selects the optimum pilot beam based upon ray tracing through the surface and is the recommended setting.

Shorter selects the shorter of the pilot X or Y phase radii.

Longer selects the longer of the pilot X or Y phase radii.

X and **Y** explicitly choose the pilot X or Y phase radii.

Plane selects a pilot beam with a waist at the surface and infinite phase radius.

User allows specification of the X and Y radius in lens units. Use the value of zero for infinite radius.

Coating

The coatings options are set in the Coating section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor.
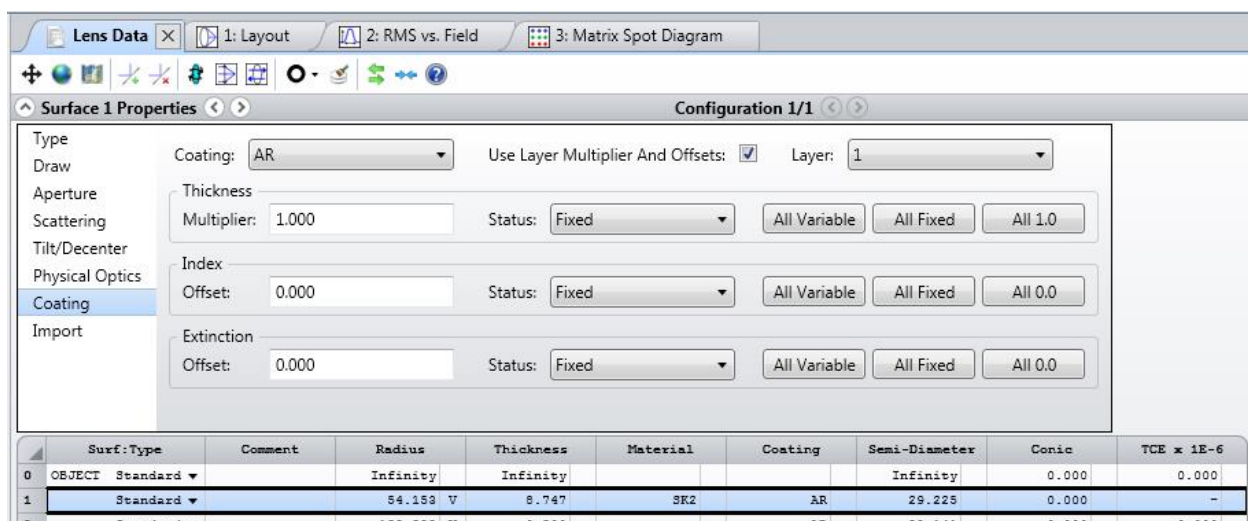
The Coating section allows selection of the optical coating to be applied to the surface. For information on defining coatings, see "Defining coatings in Zemax" and "Optimizing coatings with Zemax".

This section supports an additional capability to individually modify the thickness, index, or extinction of any layer of a defined coating applied to a particular surface without changing the underlying definition of the coating. Individual layer thicknesses may be scaled by a "coating multiplier", a dimensionless parameter. The index and extinction coefficients may be offset by a dimensionless value. Note the index offsets are applied uniformly at all wavelengths; no dispersion in the offset is supported. These parameters may be unique for every layer and every surface, even if the same coating is applied to more than one surface. Encrypted coatings do not support this feature.

Coating Settings:

Use Layer Multipliers If unchecked, all coating layer multipliers are disabled for this surface. The coating thickness will be as defined by the coating data file. If checked, the coating multipliers are considered.

Layer Chooses the particular layer to edit or review the coating multiplier for.

Multiplier The dimensionless multiplier for the selected layer.

Status The multiplier may be a fixed value, a variable for optimization, or a value picked up from a prior layer.

Index Offset The dimensionless offset of the real part of the coating index for the selected layer.

Index Status The index offset may be a fixed value, a variable for optimization, or a value picked up from a prior layer.

Extinction Offset The dimensionless offset of the imaginary part of the coating index for the selected layer.

Extinction Status The extinction may be a fixed value, a variable for optimization, or a value picked up from a prior layer.

The buttons provide a fast way of setting all multipliers and offsets to fixed or variable status or resetting the multipliers and offsets to the default unperturbed values.

Import

The option to import an extra data file is in the Import section of the Surface Properties window. The Surface Properties can be reached by clicking the down arrow in the Surface Properties bar above the Lens Data Editor.
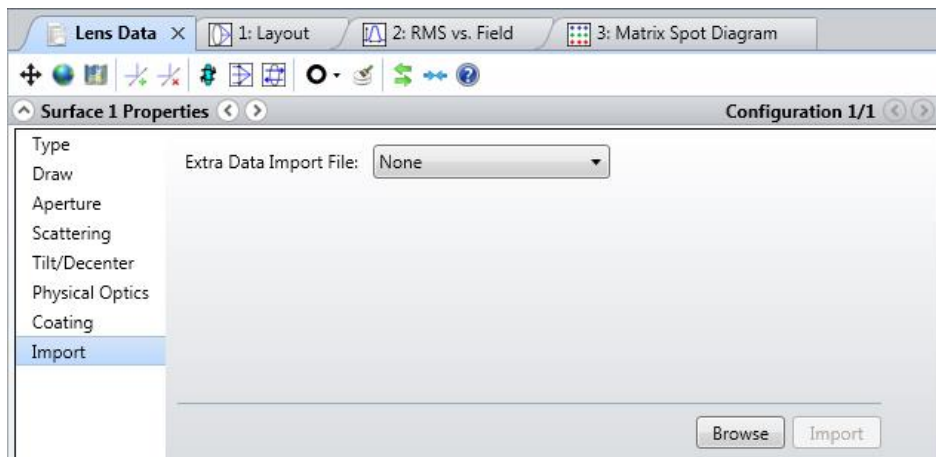
The import tool is used to load extra data values for extra data surfaces from a file rather than by typing the numbers in directly. Data files must end in the extension .DAT. Numerical data must be in the file exactly as it appears in the extra data spreadsheet. The format of the file is a single column of free-format numbers, and the file must end in the DAT extension. Data for grid sag and phase surfaces may also be imported using this tool. For a description of the proper file format, see "Importing grid data". Initially, the folder Zemax will look for DAT files is the <data> folder (see "Folders"). However, after any import or data output operation, the most recently used folder will be used.

Non-Sequential Component Editor

The Non-Sequential Component Editor button is available in the Editors section of the Setup tab. The Non-Sequential Component Editor is used in Non-Sequential UI Mode for inserting and deleting objects, selecting object types and properties, and setting solves and variables.

The following screenshot was taken with the Zemax non-sequential miscellaneous sample file "Demo 1.ZMX":

	Object Type	Comment	Ref Object	Inside Of	X Position	Y Position	Z Position	Tilt About X	Tilt About Y	Tilt About Z	Material	Front R	Z Length	Back R
1	Cylinder Pipe ▼		0	0	0.000	0.000	0.000	0.000	0.000	0.000	MIRROR	25.000	150.000	25.000
2	Rectangular Volume ▼		0	0	0.000	0.000	25.000	35.000	0.000	0.000	BK7	5.000	5.000	12.000
3	Sphere ▼		0	0	0.000	3.000	100.000	0.000	0.000	0.000	MIRROR	6.000	1	
4	Standard Lens ▼		0	0	0.000	0.000	50.000	0.000	0.000	0.000	BK7	45.000	0.000	20.000
5	Polygon Object ▼	PRISM45.POB	0	0	0.000	18.000	20.000	-170.000	0.000	0.000	SF1	5.000	1	
6	Polygon Object ▼	DOVE.POB	0	0	0.000	-10.000	110.000	-45.000	0.000	0.000	BK7	5.000	1	
7	Annulus ▼		0	0	0.000	0.000	0.000	0.000	0.000	0.000	MIRROR	25.000	25.000	15.000

The NSC Editor is only used by the Non-Sequential Components surface type supported by Zemax-Professional and Zemax-Premium. The editor is very similar to the lens data spreadsheet editor. The editor can only be activated if a Non-Sequential Components surface is present in the Lens Data Editor (See Non-Sequential Objects, Non-Sequential Sources, and Non-Sequential Detectors), or if the program mode is set to Non-Sequential Mode via the Setup tab. For information on changing the program mode, see "Program Mode".

For NSC ray tracing without ports, the NSC Editor can be used to enter almost all the needed system data (aside from wavelengths, glass catalogs, and coatings) and the Lens Data Editor is not needed. For NSC ray tracing with ports, both the Lens Data Editor and NSC Editor are needed to describe the system. Please see the reference file called "Non-Sequential Overview" for more information.

Also see the reference file called "Non-Sequential Overview" for more information on refraction and reflection from NSC objects, diffraction from NSC objects, and polarization and thin film coatings in NSC.

Object Placement

The NSC Editor allows specification of object placement. The conventions and restrictions on placing objects in the NSC group is critically important. Objects may be placed anywhere in 3D space; and objects may be placed with respect to any other object. Objects may also be placed entirely inside of other objects, or may be placed adjacent to other objects.

See Non-Sequential Objects, Non-Sequential Sources, and Non-Sequential Detectors.

Note that this section is repeated in the reference file called "Non-Sequential Overview".

Coordinate System

Each object's position is defined by 6 parameters: the x, y, and z coordinates, and the rotation about the x, y, and z axes. Zemax first decenters in x, y, and z (decenters are orthogonal so the order does not matter). Then Zemax tilts about the local x axis (which rotates the y and z axes to new orientations), then tilts about the new y axis (which rotates the x and z axes), then finally tilts about the new z axis. This is the same convention as for the coordinate break surface when using order flag = 0. The conversion from object local coordinates to global coordinates can be written in equation form as

$$\begin{bmatrix} x_g \\ y_g \\ z_g \end{bmatrix} = \begin{bmatrix} x_o \\ y_o \\ z_o \end{bmatrix} + \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix},$$

where the g subscript indicates the global coordinate, o is the offset, and l is the local object coordinate. The matrix R is the rotation matrix, which relates the orientation of local and global coordinates. These equations can be written more compactly as G = O + RL, where G is the global coordinate vector, O is the offset vector, R is the rotation matrix, and L is the local coordinate vector. For more information on properties of the rotation matrix, see "Global Coordinate

Reference Surface". The order of rotations may also be reversed, see "Use Global XYZ Rotation Order" under the Type section of the Object Properties.

Reference Objects

It is often useful to reference an object's position and rotation relative to another object. This is useful especially when placing related objects in a group, and then decentering or tilting the entire group.

The object to which the coordinates are relative to is the "reference object". The default reference object is object 0, which is the vertex of the Non-Sequential Component surface. If a positive number greater than zero is specified, then the coordinates of the object are referenced to the location and rotation of the specified object. This is an "absolute" reference object.

If the reference object number is negative, then the reference object is determined by adding the current object number to the negative reference object number. This is a "relative" reference object. For example, if the reference object is -3 on object 8, the reference object will be 5 because 8 - 3 = 5. Relative reference objects are particularly useful when copying and pasting groups of objects; this is easiest if all the objects in the group use relative references to the first object in the group.

When a reference object is used, the rotation and offset matrices then become:

$$G' = O' + R'G$$
$$G' = O' + R'[O + RL]$$
$$G' = [O' + R'O] + [R'R]L$$

Any number of coordinate reference nesting is supported; so that object 9 could be placed in the coordinate frame of object 5 which in turn is placed in the frame of object 3. The only restriction is that the reference object must precede in the object list the object whose coordinates are being referenced. To modify the reference object without the need to manually recompute the object coordinates and angles, see the "Modify Reference Objects".

Placement Inside, Adjacent, Overlapping

Placing sources within other objects, and the use of absolute and relative values for the "Inside Of" column are described in "Placing sources inside objects".

Objects can be combined to make more complex objects, by placing one object inside of, adjacent to, or overlapping another object. What determines the ray tracing properties of such a compound object depends upon the position and type of the various objects and whether or not they touch or overlap. Here the word touch means that one or more points on the boundary surface of one object is in the same location in 3D space as a point on a boundary surface of another object. Mirror objects may be placed anywhere, even in contact with or partially or fully inside any other object without restriction. Rays will always reflect from mirror surfaces back into the medium the rays had been traveling through.

Glue Distance In Lens Units

When two NSC objects are placed in contact, such as a lens touching one face of a prism, numerical roundoff will cause the ray tracing algorithm to sometimes detect a very tiny distance between the two objects. This can also occur when objects are rotated in 3D space and placed close, but not exactly, next to one another because of the finite number of digits entered in the spreadsheet editor.

The glue distance is the distance below which the objects are considered in contact. It is important that objects not be separated by distances very close to the glue distance. If it is intended that two objects be in contact, then the maximum spacing between the objects should be several times smaller than the glue distance. If it is intended that the objects be separated, then the distance between the object should be several times larger than the glue distance. Object spacings very close to the glue distance will yield inconsistent ray tracing or geometry errors. This should be avoided by adjusting either the object spacing or the glue distance.

The glue distance also determines the minimum propagation length for ray tracing. If a ray-object intersection is less than the glue distance away from the previous intercept, the intercept is ignored.

The glue distance is also used in the tolerance associated with ray tracing to general curved surfaces. Zemax will iterate until the error in the ray-surface intercept solution is less than one-fifth of the glue distance.

In the majority of cases, no adjustment should be made to the glue distance parameter. The glue distance must be no smaller than 1.0E-10 and no larger than 1.0E-03.

Nesting object limits

There is a user defined limit on the maximum number of nested objects. This defines an upper limit on how many objects can be placed inside each other. For example, if the maximum number of nested objects is 3, then object 3 may be placed inside of 2 which is placed inside of object 1. There may be any number of groups of objects each nested 3 deep in this case. The limit applies to the total nesting in any one collection of objects, however, there may be any number of such collections within the NSC group. The maximum number of nested objects is set on the Non-Sequential section of the System Explorer. Setting the nesting limit no higher than required for the system being modeled conserves memory usage, although the minimum setting is 4.

Nesting volumes

Volumes of refracting material are more complicated, because Zemax must keep track of the index of refraction through which the ray is propagating. The rule to remember is: if a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface or volume at that point.

> **If a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface or volume at that point.**

For example, if a diffraction grating lens is object #1, and a non- grating lens of the same thickness and radii made of air or glass is placed inside the first lens is object #2; then rays that strike in the zone including both objects #1 and #2 will act as though they just hit object #2.

This allows defining objects with "holes" and other compound objects. Objects may be touching at one or more faces, nested or not, or may partially overlap, to create a wide range of compound solid shapes.

Nesting surfaces

The nesting rules defined above apply to solid volumes of refractive or reflective material. Some special surfaces may also be "nested" in the sense that more than one surface may exist at the same point in space. The rules are similar to those that apply to volumes, but surfaces cannot be refracting.

The same rule applies to surfaces as does for solids: if a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface at that point.

If a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface at that point.

For surfaces, there are a few rules to consider when more than one surface exists at the ray intercept point:

1) The last surface listed will determine the properties of the surface.

2) If the last surface listed is a mirror, the ray will reflect.

3) If the last surface listed is an absorber, the ray will be absorbed.

4) If the last surface listed is neither a mirror nor an absorber, the ray will ignore the surface.

5) Surface objects may not share boundaries with volume objects unless the surface object is reflective or absorbing, or unless the volume object is listed after the surface object; in which case the volume defines the properties of the common boundary.

6) Only standard surface objects may share boundaries, support for nesting of other surface object types may be added in future versions of the software.

Summary of NSC Objects

Zemax NSC object types include ellipses, triangles, rectangles, spheres, cylinders, and other basic shapes. Complex objects such as arbitrary prisms, aspheric lenses, torics, toruses, and other optical components are also available. The reflective, refractive, and absorptive properties of these objects are determined by the material assigned to the objects.

Each NSC object type is described in the following summary list, and in greater detail in [Non-Sequential Geometry Objects](). Note these basic objects may be combined to form more complex objects. See "Object Placement" for information on placing objects inside or adjacent to one another.

If an object type is required that is not listed, please contact technical support to suggest the new object type be added to Zemax.

Summary List of NSC Objects

Annular Aspheric Lens An annular volume with an aspheric surface on one face.

Annular Axial Lens An annular lens formed by sweeping an aspheric cross section about a decentered Z axis.

Annular Volume A volume formed by two tilted, elliptical annular planar faces.

Annulus A planar ellipse with an elliptical hole in the middle.

Array This object creates a 3D parallelogram array of objects based upon a master object.

Array Ring This object creates a ring array of objects based upon a master object.

Aspheric Surface A surface with a conic substrate plus even and odd radial polynomial power aspheric terms.

Aspheric Surface 2 Identical to the Aspheric Surface, with support for more complex aperture types.

Axicon Surface An axicon formed by rotating a spherical arc around the z axis.

Biconic Lens A circular or rectangular lens with separate curvature and conic in the X and Y directions on both front and back faces.

Biconic Surface A surface with separate curvature and conic in the X and Y directions. May be partially hyperhemispheric under special circumstances.

Biconic Zernike Lens A lens with a standard front face and a biconic Zernike rear face.

Biconic Zernike Surface A surface with a biconic Zernike shape. The surface supports elliptical, rectangular, and user apertures, and the biconic and Zernike terms may be separately decentered.

Binary 1 A standard lens with a binary 1 phase profile on the front face. The binary 1 phase profile is an X-Y polynomial.

Binary 2 A standard lens with a binary 2 phase profile on the front face. The binary 2 phase profile is a radial polynomial in even powers of r. Binary 2A An even aspheric lens (front and back faces) with a binary 2 phase profile on the front face. The binary 2 phase profile is a radial polynomial in even powers of r.

Boolean A very general object defined by a Boolean combination of other objects.

CAD Assembly: Autodesk Inventor A dynamic link to an Autodesk Inventor assembly.

CAD Assembly: Creo Parametric A dynamic link to a Creo Parametric assembly.

CAD Assembly: SolidWorks A dynamic link to a SolidWorks assembly.

CAD Part: Autodesk Inventor A dynamic link to an Autodesk Inventor part.

CAD Part: Creo Parametric A dynamic link to a Creo Parametric part.

CAD Part: SolidWorks A dynamic link to a SolidWorks part. 462CAD Part: STEP/IGES/SAT An object or objects imported from a CAD program in STEP, IGES, or SAT format.

CAD Part: STL A general polygon object defined in STL format, typically exported from a CAD program.

CAD Part : Zemax Part Designer An object created using the Zemax Part Designer.

CPC A compound parabolic concentrator (CPC). The CPC may be hollow or solid.

CPC - Rectangular A CPC with rectangular symmetry and separate X and Y aperture and acceptance angles.

Cone A section of a cone defined by two points, r and z, which forms a line segment, which is then rotated around the local z axis.

Cylinder Pipe A tapered cylindrical surface shape.

Cylinder Volume A tapered cylindrical volume, with end caps

Cylinder 2 Pipe A cylindrical surface shape with tilted ends

Cylinder 2 Volume A cylindrical volume, with tilted end caps.

Diffraction Grating A standard lens with a diffraction grating with constant or variable line spacing on the front face.

Dual BEF Surface An idealized dual brightness enhancement film surface

Ellipse A planar ellipse.

Elliptical Volume An elliptical volume (or shell). The shape is a tapered cylinder with an elliptical cross section.

Even Asphere Lens A rotationally symmetric lens with up to 16th order aspheres on both the front and back faces.

Extended Odd Asphere Lens A lens with extended odd asphere surfaces on the front and back faces.

Extended Polynomial Lens A lens with extended polynomial surfaces on front and back faces.

Extended Polynomial Surface A surface with an extended polynomial shape.

Extruded A solid volume formed by extruding a User Defined Aperture.

Faceted Surface A standard surface faceted into flat polygons.

Freeform Z A solid lens or shell defined by rotating a spline curve around the Z axis. The spline curve is defined by points along the Z axis.

Fresnel 1 A true Fresnel lens, either radial or cylindrical, defined in terms of a number of construction parameters.

Fresnel 2 An idealized Fresnel lens object having a circular or rectangular shape, with one face an ideal Fresnel lens with either radial or cylindrical polynomial aspheric terms over a base conic asphere.

Grid Sag Lens A lens with a standard front face and grid sag rear face.

Grid Sag Surface A surface defined by a table of sag points.

Hexagonal Lenslet Array A lenslet array with hexagonal symmetry.

Hologram Lens A circular or rectangular solid with an optically fabricated hologram on the front face.

Hologram Surface A circular or user defined plane, conic, spherical or aspherical surface with an optically fabricated hologram on the face.

Jones Matrix An elliptical flat object which supports 8 ABCD parameters which can be use to model neutral density filters, polarizers, rotators, and other arbitrary Jones Matrix type objects.

Lenslet Array 1 A rectangular shaped lens array object with a curved back face. This object may be diffractive.

Lenslet Array 2 A rectangular shaped lens array object with curved front and back faces.

Micro Electro Mechanical System (MEMS) An array of flat faces (intended to be made of material "MIRROR") where the individual faces may be tipped about any of 3 angles. The faces may be addressed either by rows, by columns, or by individual pixels.

Null Object A non-existent object. This is the default when new objects are inserted into the NSC Editor, and is handy to use as a place holder.

Odd Asphere Lens A rotationally symmetric lens with up to 12th order odd and even power aspheres on both the front and back faces.

Paraxial Lens An idealized planar surface that acts like a thin lens with separable powers.

Polygon Object A user defined general polygon object which may be open or closed, and may have refractive, reflective, or absorptive faces, or a mixture of these types of faces.

Ray Rotator An imaginary component that rotates incident rays in two directions.

Rectangular Corner A corner composed of 3 rectangular surfaces.

Rectangle A rectangle defined by a width and height.

Rectangular Pipe A four sided box.

Rectangular Pipe Grating A four sided box with a grating on the side faces.

Rectangular Roof A roof composed of two rectangular surfaces meeting at an angle.

Rectangular Torus Surface A section of a rectangular torus which defines a surface; open at the ends.

Rectangular Torus Volume A section of a rectangular torus with closed ends to define a volume.

Rectangular Volume A six sided box enclosing a volume.

Rectangular Volume Grating A six sided box with a diffraction grating on side faces.

ReverseRadianceTM Detector A detector used with the ReverseRadianceTM feature.

ReverseRadianceTM Target A target object used with the ReverseRadianceTM feature.

Slide A color RGB file, in either BMP, JPG, or PNG format. The slide may be scaled to any size and aspect ratio, and may be used to filter a source to create colored image sources.

Sphere A sphere.

Standard Lens A 5 surface lens object with two Standard faces, two flat edges, and a cylindrical outer edge.

Standard Surface A surface with the same shape as the Zemax Standard surface (a plane, sphere, conic asphere, or hyperhemisphere).

Swept Object A surface or volume formed by sweeping the cross section of another object around an arbitrary axis through an angle.

Tabulated Faceted Radial A faceted object generated from a table of coordinates rotated about the local z axis. This object type can be used to define surfaces or volumes.

Tabulated Faceted Toroid A faceted object generated from a table of coordinates rotated about an axis parallel to the local y axis. This object type can be used to define surfaces which are cylindrical or toroidal.

Tabulated Fresnel Radial A smooth Fresnel lens generated from a table of coordinates. This is very similar to the Tabulated Faceted Radial object except the surfaces are smooth, not faceted. See also the Fresnel object.

Toroidal Hologram A circular, elliptical, or rectangular lens with aspheric toroidal surfaces on the front and back sides, and a hologram on the front face.

Toroidal Lens A circular, elliptical, or rectangular lens with aspheric toroidal surfaces on the front and back sides.

Toroidal Surface A rectangular surface with an aspheric toroidal shape.

Torus Surface A section of a torus which defines a surface; open at the ends.

Torus Volume A section of a volume torus; like a surface but with end caps to enclose a volume.

Triangular Corner A corner composed of 3 triangles.

Triangle A triangle defined by 3 points in a plane.

User Defined Object An object defined by a user provided DLL.

Wolter Surface A radially symmetric shell object used in x-ray telescopes.

Zernike Surface A surface defined by radial aspheric and standard Zernike polynomial sag terms.

NSC Editor Toolbar

The Non-Sequential Component Editor Toolbar is available at the top of the Non-Sequential Component Editor tab. This provides quick access to tools for the Non-Sequential Component Editor.
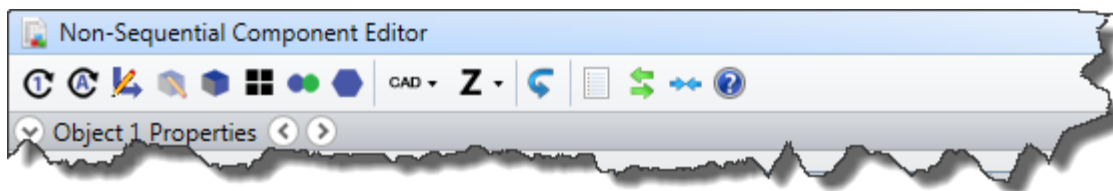
The buttons in the Non-Sequential Component Editor Toolbar are listed below.

Reload Object

Reload the data file that defines the current object. Only certain objects use data files.

This reloads and recreates an object that was defined by an external data file that was changed since the lens file was last loaded. Typically this option is used to refresh a POB or STL object if another application has modified the defining data file.

Reload All Objects

Reload the data files for all data-based objects.

This reloads and recreates all objects listed in the editor. Also see "Reload Object".

Modify Reference Object

This tool alters the reference object of any single object or range of objects. The position and rotation angle for the objects are modified to retain their current position and orientation. For more information on reference objects, see "Reference objects".

This tool works by computing the x, y, and z position and the tilt about the x, y, and z axes relative to the desired reference object to maintain each objects global position. Because of the finite precision of the computer, some small amount of error may be induced from multiple changes in the reference position.
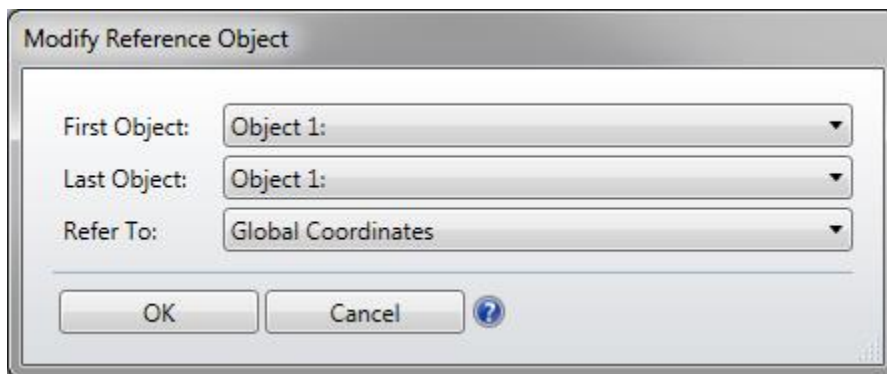
Settings:

First/Last Object The first and last object defining the range of objects to be modified. All objects within the specified range will be modified.

Refer To Defines the new reference object. All the objects within the range defined by the first and last object settings will be modified to use the new reference object. The selected reference object must precede the first object selected.

Edit Object

If the object is defined by a file, such as a Polygon Object (POB), then this menu option will be available. If selected, a text editor will be invoked so that the defining data file may be edited. After editing and saving the file; choose Reload Object to update the new information in Zemax.

For example, the following screenshot shows the Edit Object button is enabled when the polygon object is selected. The following screenshot was taken with the Zemax non-sequential miscellaneous sample file "Demo 1.ZMX" in the non-sequential program mode:

The following text editor is invoked after clicking the Edit Object button:

Replicate Object

Replicate the selected object on x, y, and z arrays.

This feature makes multiple copies of an existing object. Each copy is offset by a different amount to create an array of identical objects. The original object is placed at the center of the array, and all replicated objects are referenced to the original object's position and orientation. A generally superior alternative to this tool is the Array Object, see also "Array".
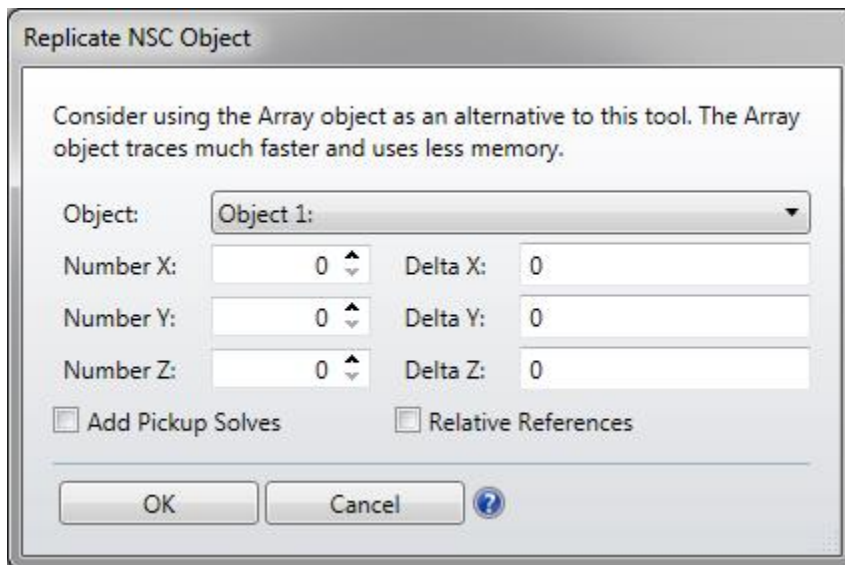
Settings:

Object The object to be replicated.

Number X, Y, Z The total number of objects in the array in the X, Y, and X directions.

Delta X, Y, Z The spacing interval along each direction.

Add Pickup Solves If checked, all replicated objects will pickup parameter data from the original object.

Relative References If checked, all replicated objects will use relative references to the original object, otherwise, absolute references are used See "Reference objects".

Combine Objects

Perform Boolean operations such as AND, OR, and XOR on any two objects.

The combine objects tool works by applying Boolean operations on the two selected objects to "trim" the overlapping regions to form a new single object. The new object is then exported in the selected CAD format, such as IGES or STEP. The new object may then optionally be imported for ray tracing. For more information on imported CAD objects, see "CAD Part: STEP/IGES/SAT". When exporting in STL format, Zemax imports the new object using the STL object type. For a description of the STL object see "CAD Part: STL".

The Boolean operations are performed by converting each component object into a NURBS based representation, and then a series of Boolean trimming and combining operations are performed to yield the resulting object. For this reason, Boolean objects have the same considerations that imported CAD objects do; see "Comments about imported CAD objects" for a discussion. Some loss of precision is possible when converting from Zemax native, high precision representations to NURBS representations; this is not a limitation of Zemax but is inherent in the NURBS representation of arbitrary surfaces. Generally, precision can be increased as desired using the Spline and Tolerance parameters described above. When extreme optical precision is required, the user should verify that the Boolean object traces to a suitable accuracy.

When the maximum possible ray trace accuracy is required, an alternative to combining objects is to create a User Defined Object. For more information, see "User Defined Object".
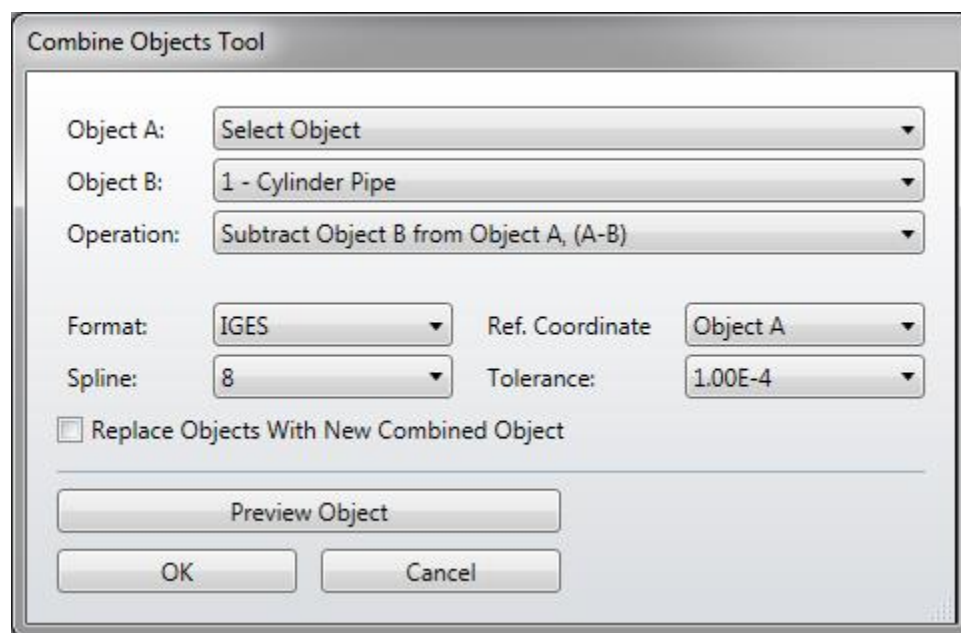
Settings:

Object A, B The first and second objects to be combined.

Operation Defines the Boolean operation to perform on the selected objects.

Intersection of objects: The region contained by both objects.

Union of objects: The region contained by either object.

Either, But not both objects: The region contained by either but not both objects.

Subtract Object B from Object A: The part of A that is not part of B.

Subtract Object A from Object B: The part of B that is not part of A.

Format The file format to use to save the resulting object.

Spline This setting defines the number of spline points to use for each aspheric face. A larger number of spline points generally produces more accurate results at the expense of larger files, slower ray tracing, and slower object creation. This setting has no effect on objects which are not aspheric.

Ref. Coordinate This setting defines the coordinate system used by the combined object. If Global is selected, the object will be position relative to the origin of the NSC system. If Object A or B is selected, then the object will be positioned relative to the local coordinate frame of Object A or B respectively.

Tolerance The dimensional tolerance in lens units. Some formats use a tolerance parameter to define the accuracy of the representation. Smaller tolerance values generally produce more accurate results at the expense of larger files, slower ray tracing, and slower object creation.

Replace Objects With New Combined Object If checked, once the object is created, the selected objects A nd B will be replaced with the newly created object. If not checked, the object will be created but will not automatically be imported to the NSC Editor.

Preview Object If pressed, this button will present a Shaded Model display of the object that would be created with the specified settings. This Shaded Model display may be rotated and zoomed to visually verify the correct shape will be created.

Create Polygon Object

Creates polygon objects of common shapes that can be used as detector objects.

Creating simple shapes as POB files is generally straightforward using a text editor. However it is often useful to make a detector object with a large number of facets, such as a cube with hundreds of facets on each side. When the object has many facets, it is easier to create the POB using this tool.

To create the POB file, choose the object type from the list, set the appropriate parameters, enter the name of the file to create, and then click OK. Note the file name should not include a path, but should include the extension POB. The tool will automatically place the POB file in the <objects>\Polygon Objects folder (see "Folders), ready to use by the NSC editor.

For information on POB objects, see "Polygon Object". For information on using objects as detectors, see "Objects as detectors". For information on the POB format, see "Defining Polygon Objects". To create objects using Boolean operations between existing Zemax objects, see "Combine Objects".

Settings:

Object Type The type object to be created.

Parameters The data specific to each object type.

Output Name The name of the POB file to create.

Various tools for handling objects created in mechanical CAD packages.

CAD

The following tools are listed in the drop-down menu of the CAD toolbar button. The availability of the feature depends on the objects in the NSC Editor.
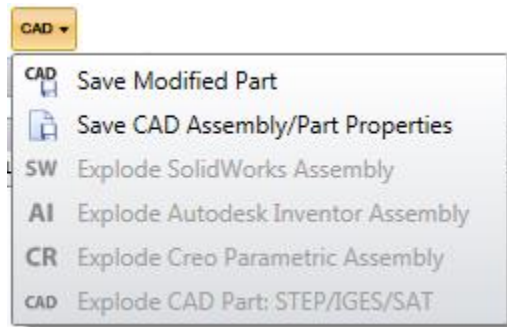
Save Modified Part

This allows an object that was originally created by a specific external program to be saved back into the format of the creating program. The specific programs that are supported by this option are Autodesk Inventor (for the "CAD Part: Autodesk Inventor" object), Creo Parametric (for the "CAD Part: Creo Parametric" object), SolidWorks (for the "CAD Part: SolidWorks" object), and Zemax Part Designer (for the "CAD Part: Zemax Part Designer" object).

Generally, saving the object back into the format of the creating program is only necessary if the original object was subsequently modified in Zemax. Thus, for Autodesk Inventor, Creo Parametric, and SolidWorks, this option is only available for part files (*.IPT, *.PRT, and *.SLDPRT files, respectively). The modified object may be saved into the same file name as the original object, thus overwriting the original file, or into a new file.
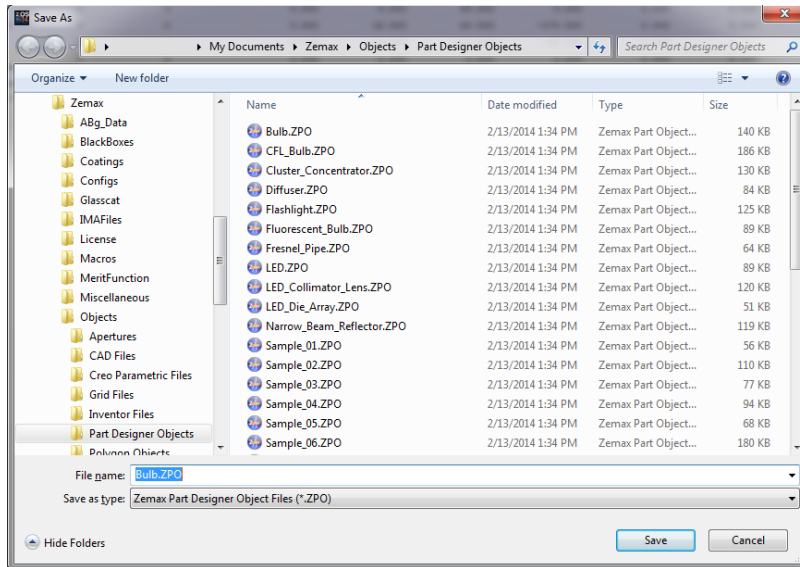
Save CAD Assembly/Part Properties

This allows optical properties assigned in Zemax to an object that was originally created by a specific external program to be saved back to the file in the format of the creating program. The specific programs that are supported by this option are Autodesk Inventor (for the "CAD Assembly: Autodesk Inventor" and "CAD Part: Autodesk Inventor" objects), Creo Parametric (for the "CAD Assembly: Creo Parametric" and "CAD Part: Creo Parametric" objects), and SolidWorks (for the "CAD Assembly: SolidWorks" and "CAD Part: SolidWorks" objects).

In Autodesk Inventor optical properties are saved under the Custom tab of the iProperties dialog box. In Creo Parametric optical properties are saved as Annotation Notes, which may be found under the Model Tree for the part or assembly (as long as Display Annotations is checked under the Model Tree Filters menu). In SolidWorks optical properties are saved under the Configuration Specific tab of the File...Properties menu. In Autodesk Inventor and SolidWorks, the name of the optical property starts with the word "Zemax", while in Creo Parametric the name of the Annotation Note starts with the letter "Z" (in this latter case the full property is not given by the note name but by the note text).

The information saved in the creating program will have no meaning to that program, but is provided only for reference. However, if the same file is subsequently loaded into another Zemax design, the property information from that file may be read in by Zemax, and used to assign optical properties to that object in the new Zemax design.

Explode SolidWorks Assembly

This tool will explode an assembly file created in SolidWorks into its individual, constituent parts. The tool is functionally identical to the one described in "Explode SolidWorks Assembly" and accessible from the File Tab. This tool can only be used to explode assembly files that are already loaded into the Non-Sequential Component Editor (NSCE) via the CAD Assembly: SolidWorks object (see "CAD Assembly: SolidWorks"). In addition, once the assembly has been exploded, each of the part files will be loaded into Zemax as a CAD object (see "CAD Part: STEP/IGES/SAT").

If the "Delete CAD Assembly: SolidWorks" checkbox is selected, then the parent assembly file will be deleted from the NSCE, leaving only the individual part files (each part represented by an imported CAD object) in the editor. Otherwise the parent assembly file will remain in the editor, with its "Explode?" flag set to 1. However, in this case the parent assembly file will be set not to draw (see "Draw") and for rays to ignore the object (see "Type"). Thus, only the constituent parts will affect performance of design.

Once the part files have been loaded into Zemax, each of them may be assigned unique optical properties, just as with any built-in object. Thus, this tool allows an assembly to be appropriately characterized in Zemax, should the different parts of the assembly have different properties.

Since each of the part files are loaded into Zemax as imported CAD objects, they cannot be modified inside of Zemax. If modification of the parts is required, each part should be loaded into Zemax via the CAD Part: SolidWorks object (see "CAD Part: SolidWorks"), rather than loading in the assembly file and then exploding it.

Explode Autodesk Inventor Assembly

This tool will explode an assembly file created in Autodesk Inventor (2013 or beyond) into its individual, constituent parts. The tool is functionally identical to the one described in "Explode Autodesk Inventor Assembly" and accessible from the File Tab. However, this tool can only be used to explode assembly files that are already loaded into the Non-Sequential Component Editor (NSCE) via the CAD Assembly: Autodesk Inventor object (see "CAD Assembly: Autodesk Inventor"). In addition, once the assembly has been exploded, each of the part files will be loaded into Zemax as a CAD object (see "CAD Part: STEP/IGES/SAT").

If the "Delete CAD Assembly: Autodesk Inventor" checkbox is selected, then the parent assembly file will be deleted from the NSCE, leaving only the individual part files (each part represented by an imported CAD object) in the editor. Otherwise the parent assembly file will remain in the editor, with its "Explode?" flag set to 1. However, in this case the parent assembly file will be set not to draw (see "Draw tab") and for rays to ignore the object (see "Type tab"). Thus, only the constituent parts will affect performance of design.

Once the part files have been loaded into Zemax, each of them may be assigned unique optical properties, just as with any built-in object. Thus, this tool allows an assembly to be appropriately characterized in Zemax, should the different parts of the assembly have different properties.

Since each of the part files are loaded into Zemax as imported CAD objects, they cannot be modified inside of Zemax. If modification of the parts is required, each part should be loaded into Zemax via the CAD Part: Autodesk Inventor object (see "CAD Part: AutoDesk Inventor"), rather than loading in the assembly file and then exploding it.

Explode Creo Parametric Assembly

This tool will explode an assembly file created in Creo Parametric (2.0 or beyond) into its individual, constituent parts. The tool is functionally identical to the one described in "Explode Creo Parametric Assembly" and accessible from the File Tab. However, this tool can only be used to explode assembly files that are already loaded into the Non-Sequential Component Editor (NSCE) via the CAD Assembly: Creo Parametric object (see "CAD Assembly: Creo Parametric"). In addition, once the assembly has been exploded, each of the part files will be loaded into Zemax as a CAD object (see "CAD Part: STEP/IGES/SAT").

If the "Delete CAD Assembly: Creo Parametric" checkbox is selected, then the parent assembly file will be deleted from the NSCE, leaving only the individual part files (each part represented by a CAD Part: STEP/IGES/SAT object) in the editor. Otherwise the parent assembly file will remain in the editor, with its "Explode?" flag set to 1. However, in this case the parent assembly file will be set not to draw (see "Draw tab") and for rays to ignore the object (see "Type tab"). Thus, only the constituent parts will affect performance of design.

Once the part files have been loaded into Zemax, each of them may be assigned unique optical properties, just as with any built-in object. Thus, this tool allows an assembly to be appropriately characterized in Zemax, should the different parts of the assembly have different properties.

Since each of the part files are loaded into Zemax as CAD Part: STEP/IGES/SAT objects, they cannot be modified inside of Zemax. If modification of the parts is required, each part should be loaded into Zemax via the CAD Part: Creo Parametric object (see "CAD Part: Creo Parametric"), rather than loading in the assembly file and then exploding it.

Explode CAD Part: STEP/IGES/SAT

This tool will explode a CAD assembly file into its individual, constituent parts. The tool is functionally identical to the one described in "Explode CAD Assembly" and accessible from the File Tab. However, this tool can only be used to explode assembly files that are already loaded into the Non-Sequential Component Editor (NSCE) via the CAD object (see "CAD Part: STEP/IGES/SAT"). In addition, once the assembly has been exploded, each of the part files will be loaded into Zemax as an imported CAD object.

If the "Delete Parent Object" checkbox is selected, then the parent assembly file will be deleted from the NSCE, leaving only the individual part files (each part represented by an imported CAD object) in the editor. Otherwise the parent assembly file will remain in the editor. However, in this case the parent assembly file will be set not to draw (see "Draw tab") and for rays to ignore the object (see "Type tab"). Thus, only the constituent parts will affect performance of design.

Once the part files have been loaded into Zemax, each of them may be assigned unique optical properties, just as with any built-in object. Thus, this tool allows an assembly to be appropriately characterized in Zemax, should the different parts of the assembly have different properties.

Freeform Z Tools

Tools specific to the Freeform-Z object.

A Freeform Z object is formed by drawing a cubic spline curve through a series of points in the YZ plane, and then sweeping this curve around the Z axis to form either a solid volume or shell. The Freeform Z object is defined by two parameters and a user selectable number of defining points:

1: The number of points. This is the number of coordinate pairs used to define the curve in the YZ plane. The minimum is 5, the maximum is 124.

2: A flag to indicate if the resulting object is a solid volume or a hollow shell. If the "Is Volume?" parameter is zero, then Zemax assumes the shape is hollow shell. No "end caps" are added in this case. If the object is a solid volume, then flat "end caps" are added if the first or last points do not have a y coordinate of zero to close the solid.

3-250: The coordinate values of the YZ curve, in pairs.

The Z coordinate of the first point must be zero. No Z value may be less than or equal to the Z value of the preceding point. The absolute value of the Y values is always taken when creating the spline curve. There is a special optimization

operand designed to provide boundary constraints on the shape of Freeform Z objects, see "FREZ" for more information. The reference point is the center of the front face of the lens. Face Numbers: Front face Face 1, back face Face 2, all other faces Face 0.

The following tools are listed in the drop-down menu of the Freeform Z toolbar button. The availability of the feature depends on the objects in the NSC Editor.
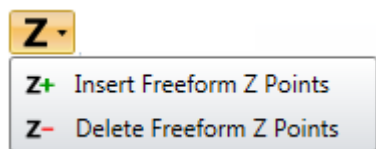
These two tools either insert or delete a single point in the list of points that define the Freeform Z object.

Insert Freeform Z Points
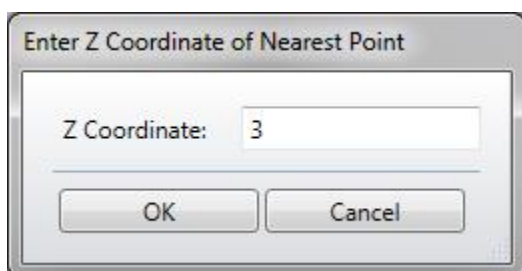
The insert tool defines the location of the point to be inserted by the Z coordinate entered on the tool dialog box. The Z coordinate must be greater than 0, less than the current maximum Z coordinate, and cannot be equal (within 1.0E-05 of the overall object length) to any existing Z coordinate. The Y coordinate is computed automatically to fit the current object shape. This tool will not insert a point if the total number of points would be more than the maximum allowed.

Delete Freeform Z Points
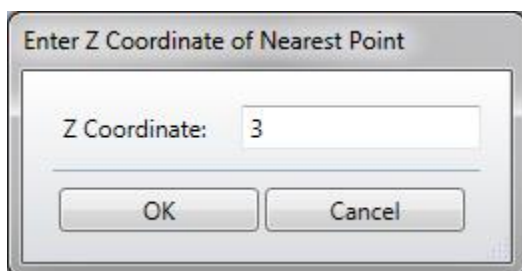
The delete tool defines the point to be deleted by the Z coordinate entered on the tool dialog box. The existing point closest to the entered Z coordinate is determined and deleted. This tool will not delete a point if the total number of remaining points would be less than 5.

Go to Object

Go to the specified Object number
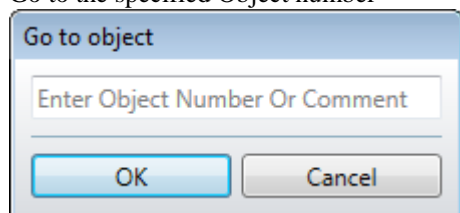
Update File Listings

Repopulates the list of available files.

Reset Column Order

Reset column order back to default settings.

Reset Column Widths

Reset column widths to a standard size.

Help

See more help on this feature.

Object Properties

Clicking the down arrow in the Object Properties bar above the NSC Editor will expand the Object Properties dialog box. The Object Properties shown in the dialog box can be changed by clicking on a different object in the NSC Editor, or by clicking the left/right arrows in the Object Properties bar. The following Object properties may be defined in this dialog box.

The following screenshot was taken with the Zemax non-sequential scattering file "Fluorescence Example.ZMX".
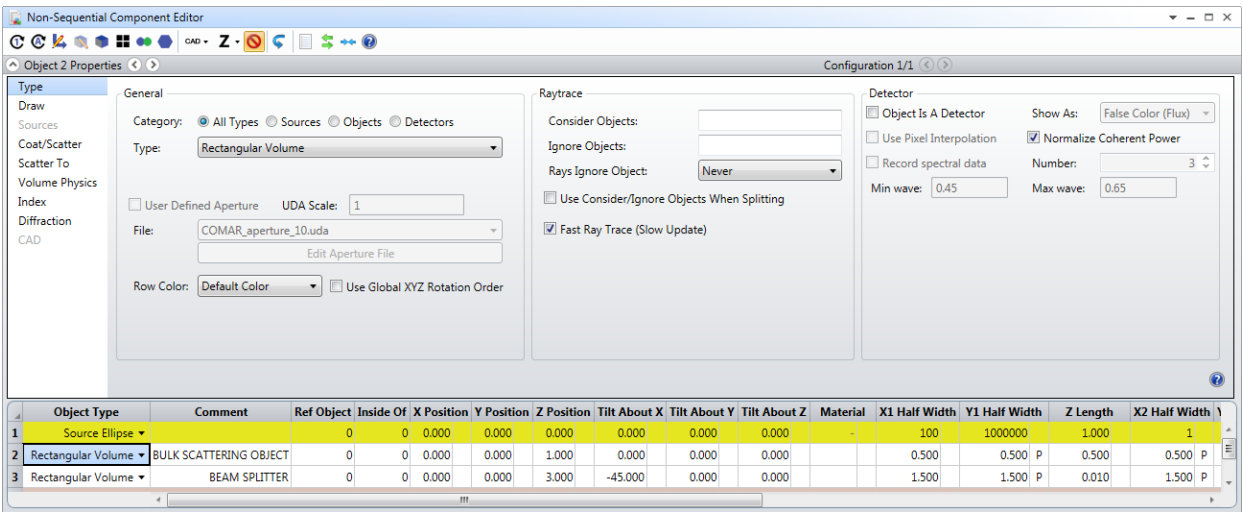
Type

Object type options are set in the Type section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.

The following screenshot was taken with the Zemax non-sequential scattering file "Fluorescence Example.ZMX".
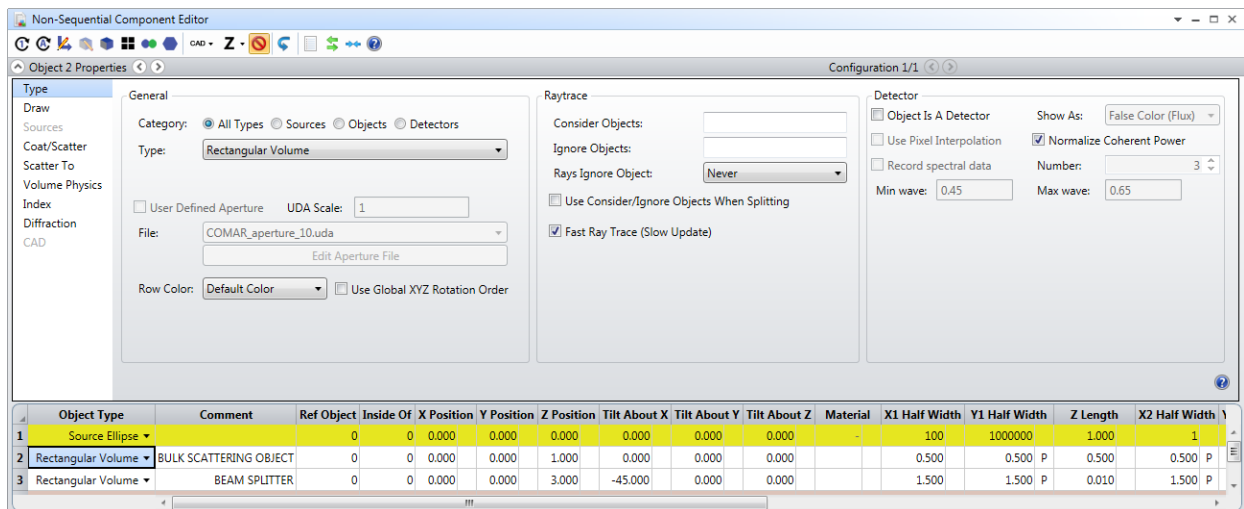
The Type section supports the following controls:

General Settings

Type Used to select the general object type, for example, sphere, ellipse, rectangle, or other type.

See Non-Sequential Objects, Non-Sequential Sources, and Non-Sequential Detectors.

Data File If the object type is defined by an external file, such as a polygon object, the file name may be selected here.

User Defined Aperture If selected, a user defined aperture (UDA) file will be used to define the extent of the object. Not all objects support this feature. User defined apertures (UDA) may be placed on some NSC surface objects. The UDA file must be placed in the <objects>\Apertures folder (see "Folders"). UDA's are defined exactly as described for sequential surfaces, see "User defined apertures and obscurations".

File If a user defined aperture is selected, the name of the aperture file may be selected from this control.

UDA Scale The UDA Scale is a dimensionless multiplier that scales the aperture defined in the UDA file.

Edit User Aperture This button will invoke a text editor to allow user editing of the selected UDA file. The UDA file needs to saved and the object reloaded to make the changes effective.

Row Color This control chooses the color of the row in the NSC Editor for the object. By default, objects made of glass, "MIRROR", or "ABSORB" material, as well as sources and detectors are color coded. Any object may use either no color, the default color, or a user defined color. The coloring of rows may be disabled, see the "Editors" section of the Project Preferences.

Use Global XYZ Rotation Order If checked, the rotation convention for object tilts is to first rotate about the X axis, then the Y axis, then the Z axis. If unchecked, the rotations are done about the Z axis, then the Y axis, then the X axis. This latter ZYX convention is what a sequential coordinate break surface does if the order flag is zero. Note that rotating using the ZYX convention is exactly the same as rotating about the X axis, then rotating about the "new" Y axis, then finally the "new" Z axis. See also, the "Coordinate system" explanation in the Object Placement section.

Ray Trace Settings

Consider Objects It is often the case that rays propagating within an NSC group follow a reasonably well defined path. For example, when several lenses are mounted in a tube, rays striking lens #5 will either hit lens #4 or lens #6 next; or possibly the tube itself. If the list of possible objects to intersect is known, and the number of objects on this list is small compared to the total number of objects; then significant ray tracing speed gains may be made by telling Zemax to only consider the subset of objects the ray may hit.

If the "Consider Objects" field is left blank (the default), then rays leaving this object may strike any object, and Zemax will use its own internal algorithms for determining which object is actually hit.

To specify the subset of objects the ray may hit when leaving an object, list the object numbers separated by a space. For example, to specify that rays leaving an object may hit objects 4, 6, or 23, enter the string "4 6 23" on the Consider Objects data field. Note the object's own object number should normally be included in the list. Zemax will automatically update this table as new objects are inserted or deleted. The maximum number of objects that may be listed is 10. If a ray may hit more than 10 different objects; leave this field blank. Use zero for the object number of the exit port.

If you are unsure as to what objects a ray leaving any object will hit, leave this field blank. If any object is listed; only the listed objects will be tested for ray intercepts! This means incorrect ray tracing results will occur if the list does not explicitly include all possible objects that may be hit by rays leaving the current object. If the ray can leave the object, and strike the same object again, the consider objects list for that object must explicitly list itself as one of the possible objects. For example, if rays striking object 5 can refract or reflect and hit object 5 again, object number 5 must be listed in the consider objects list for object 5 if a consider objects list is defined.

Note that it makes no sense to define both a "Consider Objects" and an "Ignore Objects" list. The Consider Objects list takes precedence over the Ignore Objects list if both lists are defined.

Ignore Objects The Ignore Objects list allows definition of objects that should be ignored when a ray leaves any object. A common application of this feature is for Source Objects to ignore the object from which they were defined, so that the ray does not immediately interact with object located where the ray starts.

If the "Ignore Objects" field is left blank (the default), then rays leaving this object may strike any object, and Zemax will use its own internal algorithms for determining which object is actually hit.

To specify the objects the ray may NOT hit when leaving an object, list the object numbers separated by a space. For example, to specify that rays leaving an object should ignore objects 5 and 7, enter the string "5 7" on the Ignore Objects data field. Note the object's own object number should normally not be included in the list. Zemax will automatically update this table as new objects are inserted or deleted. The maximum number of objects that may be listed is 10. Use zero for the object number of the exit port.

Note that it makes no sense to define both a "Consider Objects" and an "Ignore Objects" list. The Consider Objects list takes precedence over the Ignore Objects list if both lists are defined.

Rays Ignore Object There are three available settings for this control: Never, Always, and On Launch.

Never: The object is never ignored, and the rays will always interact with the object.

Always: Rays will never interact with the object. If an object is included for rendering or reference purposes only, ray tracing will be faster if this box is checked, as Zemax will not bother to check if the ray intercepts the object. Choosing this option can cause incorrect ray trace results if any rays could intersect the object, as rays will travel through the object as if it were not there.

On Launch: This setting means rays will ignore the object when initially launched from the source, but once the rays interact with any other object, then the object will no longer be ignored. This is very useful for objects that are close to the source location, but the source rays already consider the effects of the object. Using "On Launch" allows rays to initially ignore the close object, but then consider the object after first interacting with some other object.

Use Consider/Ignore Objects When Splitting If checked, then when rays are split the child rays will only look for intersections with the objects listed on the Consider Objects list, or only ignore objects on the Ignore Objects list. If unchecked, all possible object intersections will be checked. See "Defining paths using the Consider Objects list".

Detector Settings

Object Is A Detector If checked, then rays striking the object will increment a detector associated with the object.

Show As This control selects the color scheme used to represent objects that are detectors on the shaded model plot. Color by flux will color according to the total power incident on each area of the object. Color by irradiance will color each area by the power per area. Only objects that are detectors use this feature.

Use Pixel Interpolation If checked (the recommended setting), the energy from rays striking a pixel on the Detector Rectangle will be apportioned among the neighboring pixels. The amount of energy allocated to each pixel depends upon the position the ray strikes within the pixel. Rays striking the exact center of the pixel will have 100% of their energy allocated to that pixel. Rays striking nearer to the edge of the pixel will share their energy with pixels, if any, adjacent to the edge. Pixel Interpolation eliminates the abrupt edge response of pixels and generally yields less noisy plots and smoother optimization. Pixel interpolation is not used if the detector is using PSF mode.

Normalize Coherent Power When a ray strikes a detector, Zemax computes the real and imaginary parts of the electric field by using the intensity and phase of the ray referenced to the center of the pixel struck. The real and imaginary parts may then be summed for many rays that strike the same pixel. Zemax also sums the intensity (amplitude squared) for each pixel.

Because the phase is accounted for, some rays will constructively interfere with other rays while other rays will destructively interfere. This allows Zemax to simulate effects such as fringes in interferometers (shearing or otherwise) or interference from various orders of a diffraction grating. However, computing the coherent irradiance involves some assumptions. Physically, destructive interference means the energy would have propagated somewhere other than where the ray went. In a similar way, when constructive interference occurs, the squaring of the amplitude sum of many rays artificially and non-physically increases the energy in the beam. Zemax cannot determine where the energy went (or

came from), and therefore cannot account for conservation of energy in coherent irradiance calculations without making assumptions.

Zemax can make one of two different assumptions: either to normalize the coherent power on the detector to match the incoherent power, or to determine the coherent power using the incoherent power on each pixel and scaling by the degree of coherence of rays landing on each pixel.

If all the incoherent power from one or more sources falls on the same detector, then the Normalize Coherent Power switch should be checked on. This is the default setting. This would be case, for example, of two plane waves interfering to form a sinusoidal fringe pattern with many fringes across a detector. All of the energy is present on the detector, but the wavefront interference causes ray energy that would have landed in a dark fringe to be redirected to a neighboring bright fringe on the same detector. Conversely, when two parallel, perfectly collimated beams are combined, yielding a single bright or dark (or something in between) fringe, then the Normalize Coherent Power switch should be checked off.

See "Comments on coherent data computations".

Record Spectral Data This option is only available for the Detector Color object, and only in the Premium edition of Zemax. If checked, then the spectral distribution for all rays landing on the detector will be recorded. The distribution will be recorded into a given number of bins, as specified by the "Number" input. Only rays with wavelengths between (and including) the "Min wave" and "Max wave" inputs will be recorded; these inputs should be in microns. Once the spectral distribution is recorded, it may be plotted using the "Flux vs. Wavelength" option in the "Show As" dialog box of the Detector Viewer (see "The Detector Viewer"). If this option is selected, then the Color Rendering Index (CRI) and Color Correlated Temperature (CCT) will be calculated for the recorded distribution. The CRI and CCT values will be displayed in the Detector Viewer (when "Flux vs. Wavelength" is selected), and may be obtained in the merit function using the NSDE operand (see "Optimization - Overview").

Draw

Object draw options are set in the Draw section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.
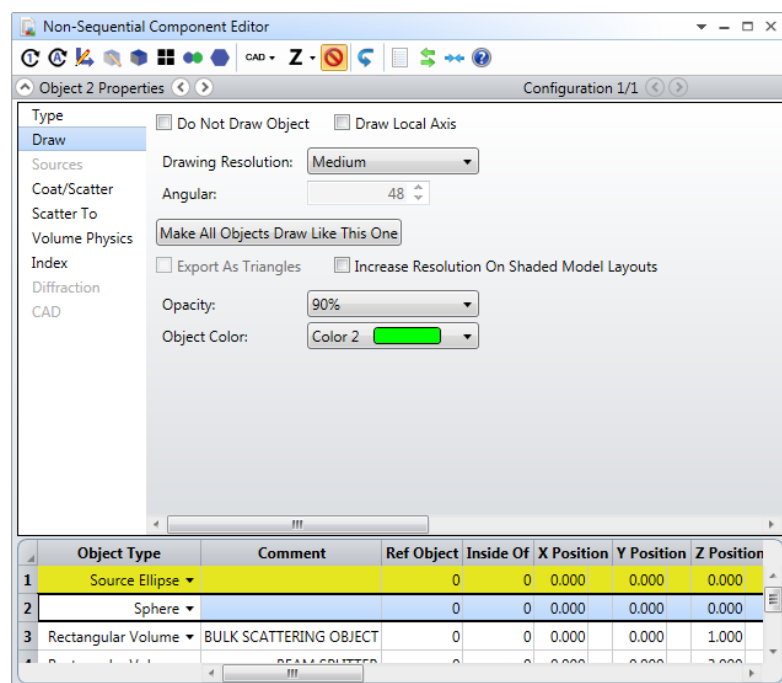
The draw section is used to set options related to drawing each NSC object. The tab supports the following controls:

Do Not Draw Object If checked, then the object will not be drawn on layout plots. Rays will still act as though the object were there. This control is useful to remove from the layout an object that encloses other objects, so the inner objects may be seen more easily.

Draw Local Axis If checked, the local x, y, and z axis will be drawn on the 3d Layout plot. The length of the x and y axis drawn is half that of the "missed ray draw distance", see "Missed Ray Draw Distance in Lens Units". The z axis is twice this length. The vertex of the coordinate indicator is at the local vertex.

Drawing Resolution Choose Standard, Medium, High, Presentation, or Custom. The options increase the resolution of the drawing at the expense of computation time. Different objects use the drawing resolution for different purposes depending upon the symmetry of the object. The Custom setting allows a user defined resolution to be specified in the edit controls below the Drawing Resolution control.

Increase Resolution On Shaded Model Plots If checked, the drawing resolution will be temporarily increased when generating Shaded Model plots. Although this produces much better looking graphics, it does take more time and memory. To save time and memory, disable this option.

Make All Objects Draw Like This One Pressing this button will copy the drawing settings to all objects.

Opacity: If the opacity is set at 100%, then the object will be rendered on the shaded model plot as a solid color, and the object may fully obscure other objects from view. If the opacity is less than 100%, then the object is partially transparent, which allows other objects to be visible through the partially opaque surface.

Object Color This control is used to select the color the object will be drawn with on the shaded model display.

Export As Triangles If checked, the object will be exported to CAD format files as a tessellation rather than a smooth object. This feature is only intended as a work around if the CAD format export feature fails to work properly. The recommended setting is not checked.

Comments about drawing resolution and bounding regions

Zemax uses the drawing resolution settings to create a "boundary region" around each object. This region is used to quickly determine if a ray comes close enough to an object to possibly intercept the object. If the ray does not cross the boundary region, Zemax assumes the ray will not hit the object. For this reason, it is possible for rays to miss an object if the drawing resolution is set too low. For these rare cases, increase the drawing resolution to ensure the ray strikes the object.

> **Do not set the drawing resolution needlessly high as this will slow down ray tracing and rendering without increasing accuracy!**

It is important to note that the drawing resolution settings are only used to render the object on the layout plots, and for some objects, to provide a "first guess" as to the location of the ray-object intercept point. For accurate ray tracing, iteration is used to find the exact intercept point. The actual surface shape of all Zemax objects are modeled exactly for ray tracing purposes; the accuracy of the ray trace is not limited by the drawing resolution, as long as the drawing resolution is sufficient to provide an adequate first guess.

Sources

Object source options are set in the Source section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.
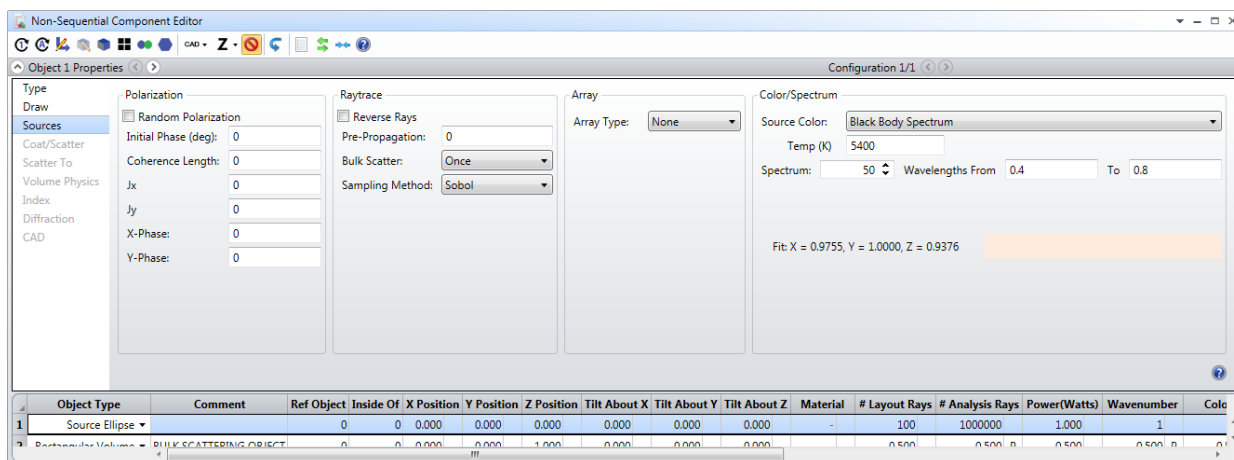
Sources section is used to define properties of source objects, including polarization state, coherence length, and initial phase, position, and direction of rays of light emanating from NSC sources. For important information see "Defining the initial polarization". The tab supports the following controls:

Polarization

Random Polarization If checked, the source will emit randomly polarized light. If unchecked, the polarization state may be defined using other controls in this section.

Initial Phase The initial phase of the ray in degrees, with 360 degrees being equal to one wave of optical path. This setting only affects coherent ray computations which depend upon the phase of the ray.

Coherence Length The length of ray propagation in lens units over which the phase is known. For details on Coherence Length effects see "Coherence length modeling".

Jx, Jy The magnitude of the electric field in the local x and y directions, respectively.

X-Phase, Y-Phase The phase in degrees of the electric field in the local x and y directions, respectively.

Ray Trace

Reverse Rays Checking this option will reverse the direction cosines of every ray. This is useful for reversing the initial direction of rays from the source. Reverse rays is done after the Pre-Propagation distance is considered, if any.

Pre-Propagation The distance in lens units the ray is propagated before beginning the actual ray trace through NSC objects. Pre propagation moves the starting point of rays forward or backward along the ray direction cosines. This

feature is useful for defining rays at one position but beginning the ray trace at a different position along the ray path; such as prior to or after an object near the source. The Pre-Propagation distance will alter the initial phase and electric field of the ray to account for the propagation length. Pre-Propagation occurs before reversing the rays if the reverse rays option is selected.

Bulk Scatter Normally, if a ray travels through an object with a bulk scattering media, the rays may scatter multiple times within the media. This is the default "Many" option. If "Once" is selected, each branch of a ray can only bulk scatter once. Note that if a ray splits before scattering, each of the child rays may scatter, since each child's branch is scattering for the first time. If "Never" is selected, then no bulk scattering will occur for rays from this source. This control is useful for modeling fluorescence. See also the "Bulk Scatter" section of the Object Properties.

Sampling Method The available options are random and Sobol. Once the parameters of a source model are defined, rays are generated randomly to model the light leaving the source. However, truly random values are not always desirable. The reason is that random numbers tend to not uniformly sample parameter space if the number of samples is small. Random numbers may group together, leaving relatively large gaps in sampling space. In practice, this means that it takes a great many rays to get sufficient sampling to produce smoothly varying results. Sobol sampling is a widely used sampling algorithm which looks qualitatively random, but is in fact a carefully selected distribution that optimally "fills in" previously unsampled space. For a discussion of the advantages and disadvantages of random and Sobol sampling, see "Random vs. Sobol sampling".

Array

Array Type This feature creates an array of identical sources, all with the same properties as the "parent" source. Depending upon the Array Type, additional parameters become available on the Source tab to define the number of array elements and the size of the array. The array types supported are:

Rectangular: This array can be used to make a 1D line or 2D array of sources with uniform spacing in the local X and Y directions. The options available include the number of sources in X and Y, and the source to-source spacing in lens units along each direction. The minimum number of sources in either direction is 1, the maximum is 2000. The sources are numbered starting from 1 in the first (x) column and first (y) row at the location of the parent object. Each subsequent source is numbered across the columns along the first row, until the number of x direction sources is reached. The next source will be placed at the next row in the first column, and then numbering continues across the columns again, until all sources are placed.

Circular: This array is a single circle of sources centered on the parent coordinate. The sources are equally spaced in angle at the specified radial coordinate in lens units. The first source is at 0 degrees on the XY plane, and the sources continue counter-clockwise (looking down the -Z axis) around the circle until all sources are placed.

Hexapolar. This array consists of equally spaced rings of sources. The first "ring" is at the location of the parent, and the first source is placed there. The second ring contains 6 sources, equally spaced in angle, starting at source number 2 at +90 degrees on the XY plane, and the sources continue counter-clockwise (looking down the -Z axis) around the circle until all 6 sources (source numbers 2-7) are placed. The third ring contains 12 sources, the fourth ring 18, and the pattern repeats until the last ring is reached. A maximum of 20 rings is supported. The spacing parameter is the radial spacing between adjacent rings.

Hexagonal. This type forms an array with hexagonal symmetry. The first "ring" contains one source at the location of the parent source. The second "ring" contains 6 sources, placed around the center source in hexagonal fashion. The third "ring" consists of 12 sources placed outside of the second ring, and the pattern repeats until the last ring is reached. A maximum of 20 rings is supported. The spacing parameter is the full height of the hexagonal region, which is the vertical spacing between sources in the same column. The numbering convention for this array starts at 1 at the bottom (-y coordinate) of the leftmost (-x coordinate) column, and then goes up the leftmost column, then starts at the bottom of the next column to the right, and proceeds up that column. The pattern repeats until the top source in the right most column is reached.

For all arrays of sources, if the total number of sources exceeds 10,000, the source objects will not be drawn. The parameter settings for the source, such as number of layout rays, number of analysis rays, and power apply to each source in the array. For example, a 3 x 3 array of 1 watt sources will produce 9 watts.

Color/Spectrum

Source Color, Spectrum, and Wavelengths From/To These settings choose the method for generating the spectral model for the source. There are a number of ways to model the spectral content of a source. Sources may be either monochromatic, or may cover some region of the visible spectrum to represent a composite color, such as orange or white. The available source color models are described below.

System Wavelength: If this option is selected, then the monochromatic wavelengths defined on the system wavelength dialog box are used. The system wavelengths are described in "Wavelengths". Once the system wavelengths are defined, the wavelength used for ray tracing is defined by the wavenumber used by the source. The wavenumber is one of the NSC Editor parameters used by all sources. See "Parameters common to all source objects" for a full description. This is the default setting for Source Color and is the only setting that uses the system wavelengths.

CIE 1931 Tristimulus XYZ: This source color model defines the color of the source using the three CIE tristimulus values commonly given the names X, Y, and Z. See "The spectrum fitting algorithm" for important information on the spectrum generated to represent the desired color. Note that although the Y value normally represents the overall brightness of the source in lumens, Zemax does not use the Y value for this purpose when defining the source color. The power or brightness of the source is set independently; see "Parameters common to all source objects" for a full description. Internally, Zemax will normalize the XYZ values to 1 lumen for purposes of computing the spectrum.

CIE 1931 Chromaticity xy: This source color model is essentially identical to the XYZ model above, except the normalized chromaticity coordinates are used instead. Zemax converts the xy values to XYZ and then follows the procedure described in "The spectrum fitting algorithm".

CIE 1931 RGB (Saturated): RGB values are converted to CIE XYZ coordinates, and the method for computing the spectrum is then the same as for CIE 1931 Tristimulus values. Note the RGB color is the fully saturated RGB color, which means colors such as grey on a 8 bit scale (128, 128, 128) are saturated to the maximum brightness (256, 256, 256) and will appear white. There is no difference in the color spectrum of grey and white, or any dark and light colors, as long as the relative color values are maintained.

Uniform Power Spectrum: This creates a uniform power spectrum over the specified wavelength range at the number of discrete wavelengths defined.

D65 White: Defines X = 0.9505, Y = 1.0000, Z = 1.0890, which is the D65 white color on computer monitors.

Color Temperature: Based upon the temperature in Kelvin, the XYZ tristimulus values are computed to yield the same color as a black body of the specified temperature, and the spectrum is computed as described for the CIE 1931 Tristimulus values. Note this is not a true black body spectrum, but a spectrum fit to yield the same color as the blackbody of that temperature.

Black Body Spectrum: Based upon the temperature in Kelvin, this yields a true blackbody spectrum over the specified wavelength range. This color model is not limited to the visible spectrum.

Spectrum File: This color model reads the wavelength values and weights from a file. For details on the file format, see "Defining a spectrum file".

CIE 1976 Chromaticity u' v': This source color model is essentially identical to the XYZ model above, except the normalized u' and v' chromaticity coordinates are used instead. Zemax converts the u' and v' values to XYZ and then follows the procedure described in "The spectrum fitting algorithm".

Note that for all Source Color settings other than System Wavelengths, Uniform Power Spectrum, Black Body Spectrum, and Spectrum File, a spectrum must be fit to represent the selected color. See "The spectrum fitting algorithm".

Random vs Sobol Sampling

Once the parameters of a source model are defined, rays are generated randomly to model the light leaving the source. This process involves generating random numbers which are scaled to represent values of the starting coordinates and direction cosines of the rays. Note the starting coordinates describe the near-field source properties, and the direction cosines represent the far-field properties.

Truly random values are not always desirable. The reason is that random numbers tend to not uniformly sample parameter space if the number of samples is small. Random numbers may group together, leaving relatively large gaps in sampling space. In practice, this means that it takes a great many rays to get sufficient sampling to produce smoothly varying results.

A solution is to use a sampling method which looks qualitatively random, but is in fact a carefully selected distribution that optimally "fills in" previously unsampled space. A widely used algorithm is a pseudo-random sequence called Sobol sampling. For a good discussion of this algorithm see Press, Flannery, Teukolsky, and Vetterling, Numerical Recipes, Cambridge Press.

The following graphs show the near-field ray patterns from two rectangular sources. The first ray pattern shown here uses random sampling. Note the distribution has no obvious periodic artifacts.
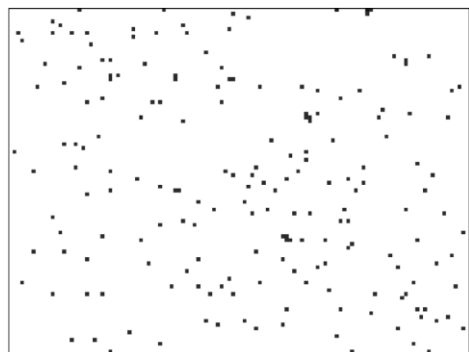
Some of the rays are very close to one another, while other relatively large areas contain no rays at all. In practice, this will yield a "noisy" irradiance distribution on a detector, since some pixels will contain one or two rays, while other pixels will contain no rays, even though the source is uniform for this sample case.

For random rays, the Signal-to-Noise Ratio (SNR) is approximately given by $N / sqrt(N) = sqrt(N)$ where N is the number of rays per pixel. For large detectors with many pixels, a very large number of rays may be required to obtain a high SNR. For example, a detector with 10,000 pixels would require roughly 100 million rays to achieve a SNR of 100, assuming the best-case scenario of a uniform source. For more practical and realistic optical systems, lower light levels would have much lower SNR.

The graph below shows the same number of rays traced to an identical detector using the Sobol pseudo-sampling grid. Note the rays are much more uniformly distributed, with relatively fewer gaps and fewer rays exceedingly close to one another. As additional rays are traced, the rays are always selected to lie between rays already traced, creating an ever finer and more uniform distribution. For reasons which are beyond the scope of this discussion, the SNR using the Sobol sampling goes as approximately N, rather than sqrt(N). For the previous example of 10,000 pixels, only about 1 million rays would need to be traced to achieve a SNR of 100, a 100X increase in efficiency.
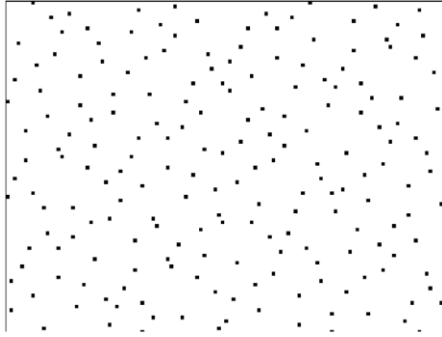
There are two disadvantages of the Sobol algorithm. First, there are sometimes subtle artifacts in the ray distribution perceptible to the human eye.

For some optical systems, if the sampling is too low, these artifacts can show up in the detected irradiance as false pseudo-periodic structures. Second, the rays generated with the Sobol algorithm are always the same.

Tracing M rays, and then tracing M more rays, will yield the exact same results as the first set of M rays; whereas tracing 2M rays would yield a different result. This is why sources using Sobol sampling as drawn on layout plots will always trace the exact same rays. The only way to get different rays is to trace at one time a larger number of rays. Nonetheless, using Sobol sampling will generally produce faster convergence with fewer rays than random sampling.

Not all source models require or support Sobol sampling. If a source model does not support Sobol sampling, this control will be greyed out. Some source models only support Sobol sampling for part of the ray generation, such as the ray starting coordinates, and not for other parts, such as the ray starting direction cosines. For any source which supports Sobol sampling (whether it be for part or all of the ray generation), this method will be the default sampling method for the source (as given in the Sources tab) when it is first added to the system.

The spectrum fitting algorithm

Zemax cannot trace a non-monochromatic "ray". Every ray traced must have a unique wavelength, so that the index of refraction, reflection, transmission, bulk absorption, diffraction, and other effects can be computed accurately. For this reason, sources that are not monochromatic must be represented by a spectrum of discrete wavelengths.

It is important to grasp that for a given composite color of light, such as light blue, the underlying monochromatic spectrum is not unique. There are an infinite number of weighted discrete spectrums that all have the same tristimulus values. When the color of a source is defined using tristimulus values or their equivalent, Zemax must compute an appropriate spectrum that represents that color. The resulting discrete wavelengths, appropriately weighted, are then traced.

There are 3 inputs that allow user control over the spectrum generation: the minimum wavelength, the maximum wavelength, and the number of wavelengths. Zemax will create a spectrum of evenly spaced monochromatic wavelengths that span the defined range. For example, if the minimum wavelength is 0.500, the maximum is 0.700, and 5 wavelengths are used, then the wavelengths chosen will be 0.500, 0.550, 0.600, 0.650, and 0.700 (all wavelengths in Zemax are in micrometers). Zemax will the compute the XYZ response at each of these wavelengths, and determine the optimal weighting at each wavelength to yield the total desired XYZ color. The resulting "best fit" tristimulus XYZ values are displayed underneath the spectrum input. Additionally, the RGB equivalent of the best fit color is displayed for reference. When using the fitting to produce a desired XYZ color, the minimum and maximum wavelengths must be between 0.380 and 0.830 micrometers, which is the practical extent of the visible range of light and the tristimulus Y response is non zero.

Note that the best fit XYZ values may differ from the desired value. This can happen if there are two few wavelengths in the spectrum, or if the wavelength range is not sufficiently broad, or if the desired XYZ values cannot be represented. In

the latter case, this is usually because the XYZ values are non-physical and do not lie within the visible region on the CIE chromaticity chart - for example, the tristimulus values (1, 1, 18) do not correspond to any visible color.

Once the spectrum is determined, Zemax randomly chooses wavelengths to trace from among the spectrum values, weighted according to the relative weights determined by the fitting process. To see the wavelengths and weights actually used in the fit, use the prescription report described in "Prescription Data".

Defining a spectrum file

Spectrum files are in text format, end in the extension *.SPCD, and are placed in the <objects>\Sources\Spectrum Files folder (see "Folders"). The files are used to describe a user-defined spectrum for source color modeling. The file format is pairs of numbers in the following format:

comment <optional>
wavelength1 weight1
wavelength2 weight2
etc...

The wavelength values are in micrometers and must be listed in ascending order. The weight values are in dimensionless relative power units. At least 3 but no more than 200 pairs of data points may be defined.

How rays are selected from the spectrum

Once a spectrum is defined, Zemax will randomly choose wavelengths from the spectrum based upon the relative weight of each wavelength. A wavelength with twice the relative weight as another wavelength will trace twice as often, and therefore twice as many rays of that wavelength will be traced. If a wavelength has a low relative weight, fewer rays of that wavelength will be traced, and this may lead to undersampling (more noise) in data associated with that wavelength.

If the source units are in Watts or Joules, every ray has equal power (or energy), and the total power of all the rays traced will exactly equal the total power of the source. However, if the source units are Lumens, this will not be precisely true. To correctly compute the true color of the detected light, Zemax must scale the ray units from Lumens to Watts. The conversion back to Lumens is done when the ray is detected. For this reason, different rays will have different power values in Watts. Since the number of rays is fixed, and the ray wavelengths are randomly chosen, the total power in Lumens will not be exactly the same value as that specified for the source. Generally speaking, if a reasonable number of rays is traced, the total Lumens will be very close to the defined source Lumens value.

Using ABg scattering with source color models

For important information on ABg coefficient scaling, see "Wavelength scaling of ABg data".

Coat/Scatter

Object coating and scattering options are set in the Coat/Scatter section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.
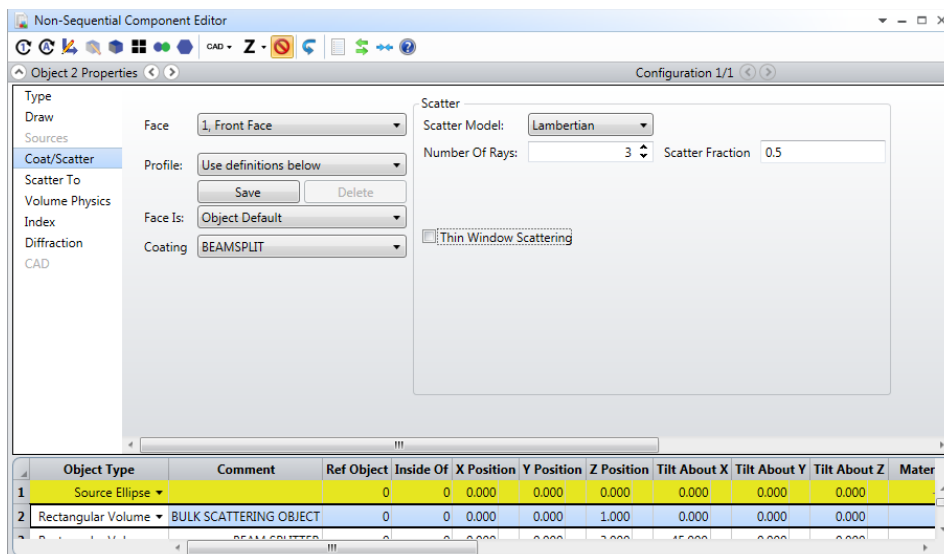
The Coat/Scatter section supports the following controls:

Face The face to which the properties will be applied. For imported CAD objects, see the "CAD" section of the Object Properties. A face is a collection of one or more surfaces on an object upon which the same optical properties are applied. For example, for a singlet lens, there are 3 faces: the front face, the back face, and all remaining faces (which include the edges and squared faces around the edges). In this case, 3 different coatings may be applied; one to each face group, if all faces of the lens are coated. Note that more than one surface may be grouped together in a single face number, and all surfaces in the same face group have the same optical properties. The valid face numbers are 0 through 50, although most objects have less than 4 faces.

Profile A profile is a collection of settings related to the thin film coatings and scattering model data applied to an object face. Suppose that many objects in an optical system are all composed of the same material and have the same coating and scattering properties. Rather than type this identical data in for each object and face; the data may be typed in once, then saved to a profile. Once saved, any other object face may use the same profile. Profiles are stored in the file whose name is specified on the Files section of the System Explorer; see "Scatter Profile" for details. If unique settings for this object are desired, then choose "Use definitions below" to edit the scattering parameters. If a profile is selected; then the coating and scattering data boxes will be disabled, as this data is defined by the selected profile. However, the settings will be displayed.

Save Saves the current scattering settings as a new profile. Pressing this button will prompt for the name of the new profile.

Delete Removes the currently selected profile from the Scatter Profile file.

Face Is This setting controls whether the face is refractive, reflective, or absorbing. "Object Default" means the face is either refractive, reflective, or absorbing, depending upon the material type defined in the NSC Editor. "Reflective" means the face is always reflective, and "Absorbing" means the face is always absorbing.

Coating The name of the coating to apply to the surfaces with the selected face number. Coatings are defined in the system coating file, see "Defining coatings in Zemax" for details.

Polarization and thin film coatings

Rays traced through non-sequential components may be done while accounting for polarization effects, or polarization may be ignored. The initial polarization state for a ray is determined by the source properties, see the "Sources" section of the Object Properties. If polarization ray tracing is being used, transmission, reflection, and absorption of optical

---

energy is accounted for at all surfaces. Bulk absorption is also accounted for. Thin film coatings significantly affect transmission and reflection properties of optical surfaces. Surfaces are initially uncoated, but coatings may be applied to surfaces or group of surfaces.

Coatings on surfaces in contact

If two surfaces are in contact, such as two 45-45-90 prisms placed so that one face of each prism is in contact with the other, then a coating may be applied "between" the surfaces in contact.
This is accomplished using the same convention described above in the section "Object Placement". The rule is: the LAST object listed in the NSC Editor determines the properties of the interface between two objects.
For example, to place a thin metal coating on the interface between two prisms arranged to form a beam splitter, the first prism object listed should have the contact surface "uncoated", while the second object listed should have the contact surface coated with the appropriate thin film coating. Rays striking this interface from either side will see the correct coating, and the ray transmission and/or reflection will be correctly computed. Note prisms modeled as POB objects can have different coatings applied to different faces, so some faces may be anti-reflection coated while others are coated with a reflective coating.

Scattering Models

Scatter models are defined in terms of a probability distribution function. When Zemax scatters a ray, a new direction of propagation is chosen. The direction is chosen using a probability function and one or more random numbers. The net effect is that if many rays are traced, the resulting scattered ray distribution would approach the probability distribution function. There are seven scattering models available: none, Lambertian, Gaussian, ABg, ABg File, BSDF, and user defined. Each available scatter model is summarized below, but see "Scattering" for the detailed technical descriptions.

No Scattering The default scatter model is "No Scattering", which means no scattering will occur. The resulting ray is called the unscattered or the "specular" ray (even if the surface is not actually reflective). The Bi-Directional Scatter Distribution Function (BSDF), which is the scattered radiance per unit incident irradiance, is zero.

Lambertian In the Lambertian scattering model, the scattered ray projection vector has equal probability anywhere in the unit circle, and the BSDF is just $1/\pi$. The scattered intensity is proportional to the cosine of the angle between the normal vector and the scattered ray angle. Note Lambertian scattering is independent of the ray incident angle. Most diffuse surfaces are nearly Lambertian. Although Lambertian is a valid option on the sequential surface dialog, rays which scatter using this model may go in any forward direction, which may cause rays to scatter at large enough angles so that they do not correctly propagate through the rest of the optical system.

Gaussian In the Gaussian scattering model, the scattering distribution is rotationally symmetric in direction cosine space, no matter what angle the specular ray makes with respect to the surface normal. The BSDF expression (see "Scattering") contains a dimensionless value σ, which determines the width of the Gaussian distribution on the projected plane. Values of σ greater than about 5.0 yield a BSDF that is nearly Lambertian. For this reason, the maximum allowed value of σ is 5.0.

ABg The ABg scattering model is a widely used method for defining the BSDF. This scattering model is generally a good model to use when the scattering is mainly due to random isotropic surface roughness, and the scale of the roughness is small compared to wavelength of light being scattered. These assumptions are generally valid for polished optical surfaces. See "Scattering" for a detailed technical description.

ABg File The ABg File scattering model allows a sum of ABg profiles to be used to define the scattering properties of a surface. The profiles to use are specified in a text file. The text file must have a .ABGF extension and must be located in the <data>\ABg_Data folder (see "Folders"). All of the profiles specified in the ABGF file must be defined in the currently loaded ABg Data File (see "ABg Data File"), and all profiles must be specified using upper case letters (regardless of capitalization of the profile name in the ABg Data File). description.

BSDF The BSDF scattering model allows the use of tabular BSDF data for defining the scattering properties of a surface. Data are provided via text files. Files must follow the BSDF Data Interchange file format described in the article entitled "BSDF Data Interchange File Format Specification", which may be found in the Zemax Knowledge Base. Zemax allows definition of separate BSDF data for reflection and refraction. If the specular ray reflects or refracts, the ray is subsequently scattered using data from the appropriate input file. For a file to be available for use in this scatter model, it must have a .BSDF extension (as indicated in the article describing the file format), and be located in the <data>\Scatterdata folder (see "Folders"). A full description of the model and its use is provided in the Knowledge Base article entitled "How to Use Tabular Data to Define the Surface Scattering Distribution" also available on the Radiant Zemax web site. description. description.

User defined scattering Completely general surface scattering may be defined via an external program called a Dynamic Link Library (DLL). Sample DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs" and "Scattering – User defined scattering" for a detailed description.

IS Scatter Catalog The IS Scatter Catalog scatter model allows the use of data measured with the Radiant Zemax IS-SA^{TM} device for defining the scattering properties of a surface. Data are provided via binary ISX files, a library of which is provided with Zemax (see "Download IS® Scatter Catalog Data").

Zemax allows definition of separate data for reflection and refraction. If the specular ray reflects or refracts, the ray is subsequently scattered using data from the appropriate input file. For a file to be available for use in this scatter model, it must have a .ISX extension and be located in the <data>\Scatterdata folder (see "Folders"). The maximum number of characters that the file name can contain (including the .ISX extension) is 60.

This model is functionally very similar to the BSDF scattering modeling described above, and as such many of the descriptions provided in the Knowledge Base article entitled "How to Use Tabular Data to Define the Surface Scattering Distribution" apply to this model as well. However, Zemax can also read spectral information from the ISX file, if such information exists. Thus, the variation of scattering with wavelength is supported in this model.

The user is allowed to specify the angular resolution with which data from the ISX file is read. Current Zemax options are 5 degrees, 2 degrees, and 1 degree, in which the value refers to the angular resolution in both the polar (radial) and azimuthal directions. Please contact Zemax technical support if you require a different resolution option and/or higher resolution.

The IS-SA may be used to measure either the front or back side of a scattering sample, or both sides. The user must then specify which side of the sample to use in defining the scattering distribution for the surface. If the user selects a side for which data are not available, Zemax will issue an error message and force selection of the other available side.

The ISX file contains the full energy distribution for light scattering from the sample (in reflection and/or refraction). As such, no degrees of freedom exist to specify an independent face property, coating or scatter fraction for any surface on which this scatter model is applied, and those controls will be greyed out when the IS Scatter Catalog scatter model is selected for a surface.

Fraction to scatter and number of scatter rays

If a scattering model other than "No scattering", "ABg", "ABg File" or "IS Scatter Catalog" is selected, the "Fraction to Scatter" must be defined. This fraction must be between 0.0 (no rays will be scattered) and 1.0 (every ray will be scattered). For the ABg and ABg File scattering models, the fraction to scatter is determined by the ABg parameters, see "Defining ABg data". For the IS Scatter Catalog scattering model, the fraction is always 1, with no user control over this input, as the total integrated scatter for this model is included explicitly in the measured data.

If ray splitting is off, the decision to scatter or not to scatter is made by the generation of a single random number between 0.0 and 1.0. If this random number is larger than the fraction to scatter, the ray will not scatter, otherwise, the ray will scatter. For example, if the fraction to scatter is 1.0, the ray always scatters. If the fraction to scatter is 0.0, the

ray will never scatter. If the fraction to scatter is 0.25, then on average one out of four rays will scatter. All of the energy of the ray follows the randomly generated scatter path. The number of scatter rays has no affect if ray splitting is off.

If ray splitting is on, then Zemax will split the specular ray into one or more scattered rays, while still possibly tracing the specular ray. The specular ray will receive a fraction of the original energy equal to (1.0 - f) where f is the fraction to scatter. The remaining energy will be divided equally among the one or more scattered rays. The number of scatter rays determines how many scatter rays will be generated. For example, if the fraction to scatter is 1.0, then the specular ray will receive zero energy and will no longer be traced; and all the energy will be divided equally among the scattered rays. If the fraction to scatter is 0.0, no scattered rays will be traced, and the specular ray retains all the original energy. If the fraction to scatter is 0.25 and the number of scatter rays is 5, then the specular ray will receive a relative energy of 0.75, and each of the 5 scattered rays will have a relative energy of 0.05. If the number of scatter rays is set to zero, then the fraction to scatter is ignored and no scattering occurs. The maximum number of scatter rays is 300, although much smaller values are typically used.

Thin Window Scattering

When this option is selected, rays will only scatter in reflection upon entering a scattering volume (i.e. when entering the window from air) and will only scatter in transmission when exiting the volume (i.e. when exiting the window into air); otherwise rays will follow the specular ray path. This option is generally used when modeling a thin scattering volume in which the scatter data were measured, i.e. in which the BSDF and/or IS Scatter Catalog scatter models are selected.

When measuring scattering in reflection or transmission from a thin volume, the reflected (BRDF) and transmitted (BTDF) measurements encompass all of the scatter paths in the volume. Since each single set of data (BRDF or BTDF) encompass all real scatter paths, all of these paths need to be consolidated when modeling scatter in Zemax. Thus, with selection of the Thin Window Scattering option, scattering only happens once in reflection and in transmission from the volume.

Scatter To

Object "Scatter To" method is set in the Scatter To section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.

A very large number of rays may need to be traced to find a relatively small number of scattered rays that strike an object of interest, such as a detector. There are two ways to improve the efficiency of the scattering analysis.

The first method is to scatter a ray according to the scatter distribution, but only trace the ray if the ray propagates towards an object of interest. This method may be implemented by defining a "Scatter To" list of objects. The Scatter To method works well for wide angle scatter (such as Lambertian scatter) and when the object of interest subtends a relatively large angle as seen from the scattering surface.

The second method is to always scatter the ray towards the object of interest, and then to normalize the energy the ray carries to account for the probability the ray would have actually scattered in that direction. This method is called "Importance Sampling". Importance Sampling is generally superior to the Scatter To method if the scatter is narrow angle or the object of interest subtends a relatively small angle as seen from the scattering surface.

The Scatter To List

The Scatter To feature speeds up scattering analysis by ignoring scattered rays which do not propagate directly towards an object of interest. The Scatter To list is very similar to the Consider Objects list (see "Defining paths using the Consider Objects list"). The Scatter To list is a string of integer object numbers separated by spaces. A scattered ray will only be traced if the ray intersects one of the objects listed or the object that scattered the ray. For example, suppose object 2 defines the Scatter To list as "3 4". If a ray scatters off object 2, it will only be traced if the ray will intersect objects 2, 3, or 4. Note that any other objects are ignored, so scattered rays will miss any object not explicitly listed (or the scattering object itself), even if that object lies directly in the path between the scatter point and the listed scatter to object.

Choosing an object on the Scatter To list does not guarantee that a scattered ray will be traced to that object. The scattered rays are generated based upon the scattering properties of the object the incident ray strikes (see "Scattering"). Zemax will generate the scattered rays, then ignore those rays that do not intersect any of the listed objects. This method does not ensure that a ray will scatter toward a listed object. However, the number of scattered rays (see "Fraction to scatter and number of scatter rays") may be made large so that it is more probable some of them will head in the desired directions.

To turn off the Scatter To feature, leave the list blank. In this case, all scattered rays will be traced. The key difference between the Scatter To list and the Consider Object list is that the Scatter To list only applies to scattered rays, while the Consider Object list applies to both specular and scattered rays.

Importance Sampling

Importance Sampling speeds up scattering analysis by always choosing scattered rays that will propagate in a desired direction. To determine the direction for the scattered ray, a target sphere is defined. The target sphere is defined by position, size, and limit. The target sphere position is defined by the selection of an object number. The target sphere will then be placed at the origin of the selected object. Any object type may be used, including Null objects. The size parameter defines the radius in lens units of the target sphere. The limit parameter is used to define the maximum solid angle in Steradians of the target sphere as seen from the scatter point. If the solid angle of the target sphere as seen from the scatter point exceeds the limit value, Importance Sampling will not be used for that scattered ray. The purpose of this feature will be discussed below.

When the ray scatters from the surface, the direction and subtended solid angle of the target sphere as seen from the scatter point are computed. A ray direction is randomly selected from within this solid angle cone with uniform probability. This will be the scattered ray direction. Note that the direction is not dependent upon the BSDF of the surface.

Because the ray direction is determined without consideration of the BSDF, the energy the ray carries must be modified to get correct ray tracing results. The amount of energy the scattered ray will carry depends upon the BSDF and TIS of the surface as follows:

$$E' = E\frac{BSDF}{TIS}\Omega\cos\theta$$

where E is the energy allocated to the scattered ray before importance sampling is applied, E' is the amount of energy in the ray traced towards the target sphere, Ω is the solid angle of the target sphere as seen from the scatter point, θ is the angle between the local normal and the scattered ray, and BSDF and TIS are the usual scattering function and total integrated scatter (see "Bi-Directional Scatter Distribution Function"). Note this energy renormalization is only accurate if the BSDF is reasonably uniform over the solid angle of the target sphere. If Ω becomes large compared to the variation of the BSDF, the energy normalization will not be accurate. For this reason, a limit value may be defined. If the target sphere is close enough to the scatter point so that Ω exceeds the limit, then Importance Sampling is not used for that scattered ray.

Up to 6 independent target spheres may be defined for each object. If one scatter ray is selected, the first target sphere is used. If two scatter rays are selected, then the first two target spheres are used, and so on up to 6 scattered rays. If more than 6 scattered rays are selected, the seventh and subsequent rays scatter according to the BSDF and Importance Sampling is not used. Energy normalization is complex and somewhat prone to statistical errors when the number of scattered rays is not equal to the number of target spheres. For the fastest, most accurate results, use a number of scattered rays that is exactly equal to the number of target spheres.

Volume Physics

The object bulk scattering method and photoluminescence settings are defined in the "Volume Physics" section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.

The following screenshot was taken with the Zemax non-sequential scattering sample file "Bulk Scatter.ZMX":
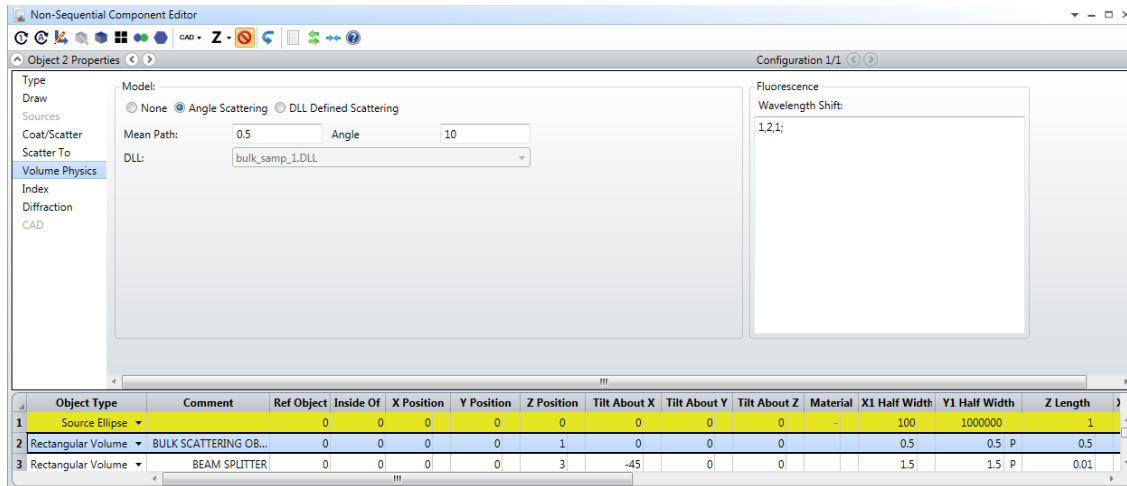
Bulk (volumetric) scattering models random scattering of rays while propagating through a solid object. Zemax supports 3 modes for the bulk scattering of a solid object: No Bulk Scattering, Angle Scattering, and DLL Defined Scattering.

None

This means that no bulk scattering is selected, and rays will propagate through the solid without scattering.

Angle Scattering

Angle scattering uses a simple model for scattering within a solid. Rays traveling a distance x within the media have an integrated probability of having been scattered given by

$$p(x) = 1.0 - e^{-\mu x}, \text{ where } \mu = \frac{1}{M},$$

and the symbol M is the mean free path in lens units. Note that as x increases, the probability that the ray has scattered asymptotically approaches 1.0. Setting this expression to a randomly selected value between 0.0 and 1.0, then solving for x, yields a randomly generated path length with the correct statistics. If this path length is greater than the distance the ray propagates to the next ray-object intersection, then no scattering occurs; otherwise, the ray scatters at the specified position along the propagation direction.

Once the position is determined, the scattering is modeled by a new ray direction being chosen. For angle scattering, a random angle is chosen so that the new ray direction lies uniformly distributed within a cone that makes an angle with respect to the current ray direction. The semi-angle of the cone is one of the parameters to the model, and this parameter should be set to between 0.0 and 180 degrees. The latter value will scatter the ray randomly in any direction. Once the ray scatters, Zemax automatically will adjust the polarization vector and randomize the phase of the scattered ray.

DLL Defined Scattering

If the angle scattering model is not sufficient, more complex bulk scattering functions may be defined via an external program called a Dynamic Link Library (DLL). Sample DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler. See also "Comments about DLLs".

Index

The object index settings are defined in the "Index" section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.

This section is used to define the index type for objects made of isotropic, birefringent, or gradient index materials.

Isotropic Types

Birefringence is ignored, and the object is modeled as an isotropic media whose index is defined by the material associated with the volume.

Birefringent Types

The birefringent type is used to define birefringent properties of solids made from uniaxial anisotropic materials, such as calcite.
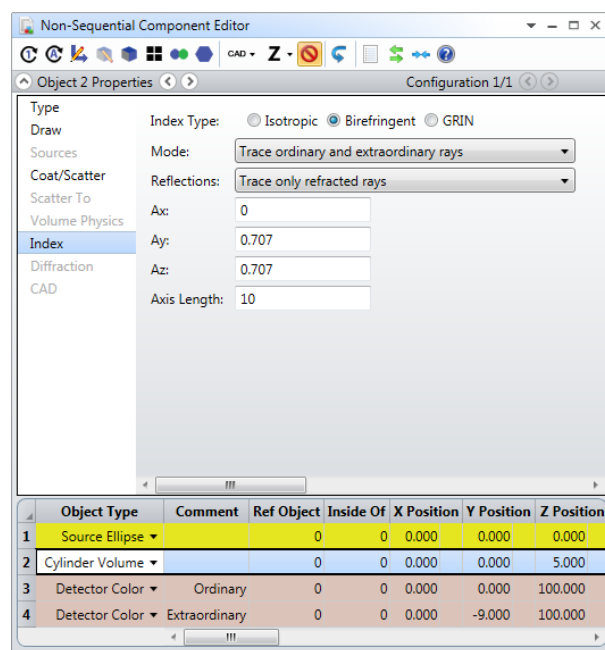
Settings:

Mode This control allows selection of which rays are actually traced. The distribution of energy does not change with this setting, and energy assigned to rays that are not traced are placed in the "lost energy (thresholds)" sum. Either ordinary or extraordinary rays, or both, may be traced. Waveplate Mode traces the energy along the path of the ordinary ray, however, the extraordinary electric field is also propagated with the ray, and the total electric field will be phase rotated by the differential index in the ordinary and extraordinary directions.

Reflections This control determines whether refracted or reflected rays, or both, will be traced. Note rays that TIR at a surface are considered reflected rays.

Ax/Ay/Az The dimensionless components of the uniaxial crystal orientation vector in object local coordinates. These coordinates define the orientation of the axis of symmetry for the media. Internally, the coordinates are normalized to a magnitude of unity. If all three terms are zero, the orientation used is (0, 0, 1).

Axis Length The length in lens units of the crystal axis drawn on layout plots. Use a value of zero to not draw the crystal axis.

Birefringence is ignored unless both polarization and ray splitting are on. If birefringence is ignored, the media acts like an isotropic media whose index is defined by the material associated with the volume. If birefringence is considered, the material associated with the object defines the "ordinary" refractive index. The "extraordinary" refractive index is defined by the material whose name is determined by appending "-E" to the ordinary material name. For example, if the object material name is CALCITE, then the extraordinary material must be named CALCITE-E. If both material names are not found in the current glass catalogs, an error is issued. If the material is defined as either a table or model glass, the extraordinary and ordinary indices are set to the same value and the media is effectively not birefringent.

When a ray strikes a boundary where either the incident or substrate media (or both) is anisotropic, as many as four separate rays are computed - two for ghost reflection and two for refraction. In each direction, there may be a separate ordinary and extraordinary ray. The ordinary ray carries the portion of the electric field that lies orthogonal to the plane that contains the direction of propagation and the crystal axis vector, while the extraordinary ray carries the electric field that lies in that same plane. If the media the rays are traveling in is isotropic (such as the ghost reflection off of a birefringent object) then the ordinary and extraordinary rays are combined into a single ray for convenience, as the energy propagates along the same path in this case.

Although there are important differences between the non-sequential and the sequential model for birefringent media in Zemax, most of the discussion about the birefringent media itself is common to both models. This discussion is found in "Birefringent In and Birefringent Out".

Birefringent media may not also be gradient index or use bulk scattering, and may not be diffractive or scatter rays at the media boundaries. Simple ray splitting is not supported for birefringent media. Sources may not be placed inside birefringent media.

GRIN Types

The GRIN type is used to define the properties of solids made of a gradient index material.
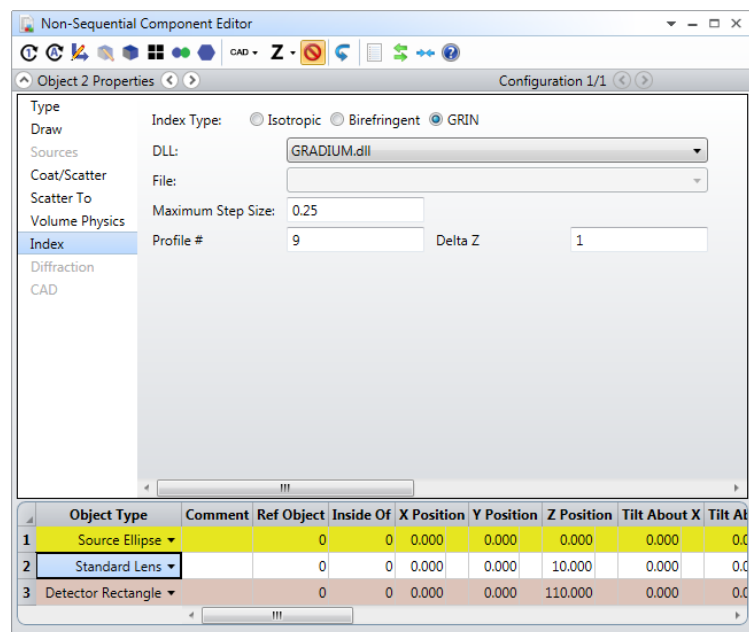
Settings:

DLL The name of the DLL to use. All gradient index materials used by non-sequential objects are defined in a separate program called a Dynamic Link Library (DLL). Numerous DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler.

Maximum Step Size The maximum step size to use during piecewise ray tracing. The maximum step size determines the trade-off between ray tracing speed and accuracy. The exact value required depends upon the rate at which the index of refraction changes and the desired accuracy of the computations. If the maximum step size is too large, the ray trace results will have large errors, and some rays will miss objects they would otherwise hit, or vice-a-versa.

Remaining controls Each DLL may use between zero and 12 user defined data values as parameters in the computation of the media properties. These values are defined by the DLL and are only used by the DLL.

To make these parameters variable or under multi-configuration control, use the NPRO operand; for details see "NPRO".

For more information on defining GRIN media, see "Defining GRIN media for non-sequential ray tracing".

Diffraction

The object diffraction settings are defined in the "Diffraction" section of the Object Properties window. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.
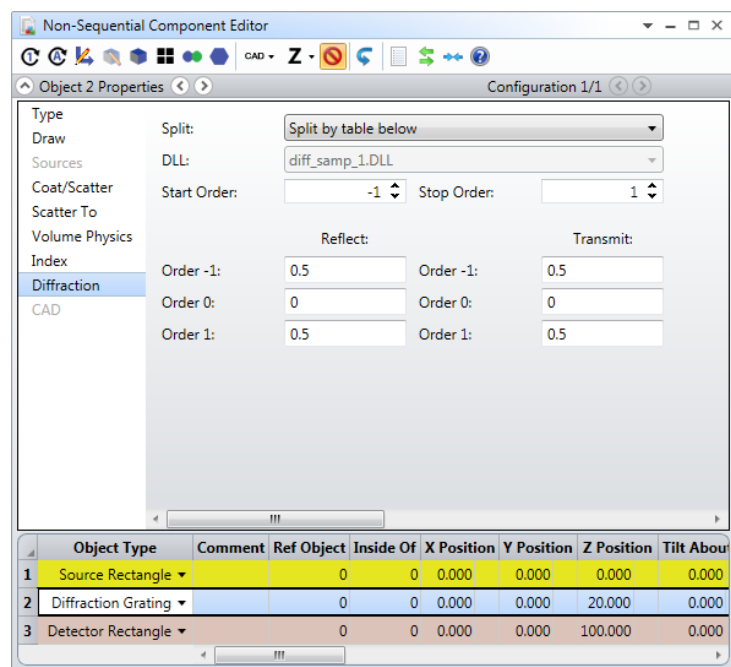
This section is used to define the properties of diffractive surfaces and how rays are split off.

Settings:

Don't split by order The rays will not split at the surface. Only the order defined by the object parameters will be traced, and all the transmitted energy goes into this one order.

Split by table below A user defined number of rays will be traced over a range of integral orders. The fraction of energy given to each order for reflected and transmitted (if any) rays is defined by the user in a table. The sum of all the defined fractions should not exceed 1.0 to properly conserve energy. The Start/Stop order are the beginning and ending order number. These numbers determine how many cells in the table are active. The order limits must be between -100 and +100, inclusive.

Split by DLL function Any diffractive surface may split rays that started at a non-sequential source, by order, to allow for simultaneous tracing of multiple diffracted orders. How many rays split, and how much energy is in each order, may be defined externally in a separate program called a Dynamic Link Library (DLL). Numerous DLLs are provided with Zemax with source code. New DLLs may be easily created with a suitable compiler.

The Start/Stop order are the beginning and ending order number. These numbers determine how many cells in the table are active; and how many times the DLL will be called to compute the output ray properties. The order limits must be between -100 and +100, inclusive. When using user defined objects in mixed sequential/non-sequential mode, no splitting is allowed, and only the start order is used to define the diffraction order used for ray tracing.

The remaining controls are used to define the parameters to be passed to the DLL. See "Creating a new Diffraction DLL". Each DLL may use between zero and 12 user defined data values as parameters in the computation of the diffractive surface properties. These values are defined by the DLL and are only used by the DLL. To make these parameters variable or under multi-configuration control, use the NPRO operand; for details see "NPRO".

CAD

The imported CAD objects, the Source Imported, the Boolean object, and the Swept object use the CAD section of the Object Properties dialog. The Object Properties can be reached by clicking the down arrow in the Object Properties bar above the NSC Editor.
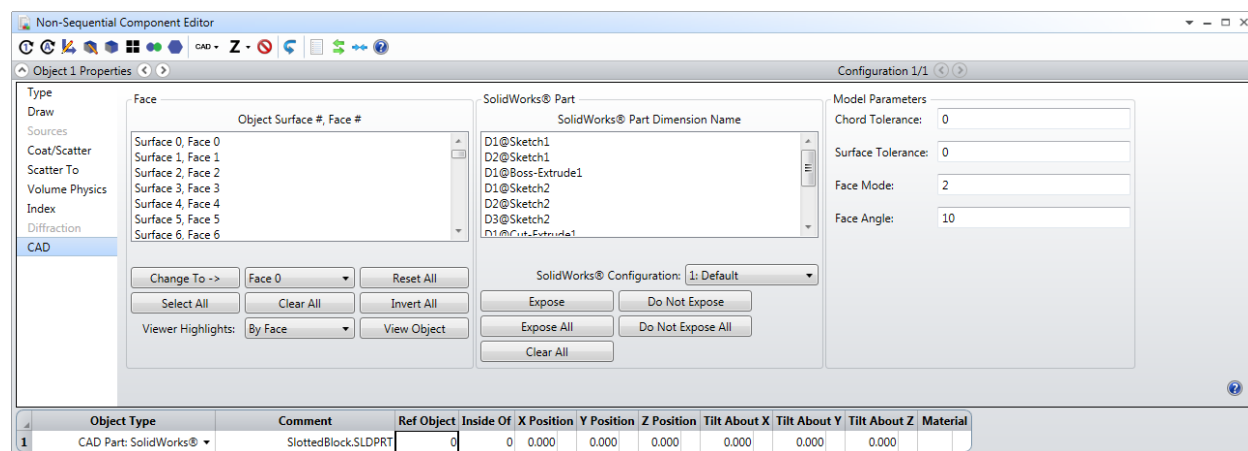
Face Settings

The controls in this section are used to assign face numbers to the various surfaces defined in an imported CAD format file. The Face number assignments are stored in a file with the same name as the imported CAD file, with the extension *.ZEN appended. In this case, "SlottedBlock.SLDPRT.ZEN". The *.ZEN file is stored in the same folder as the CAD file. If the object definition or Face Mode changes, the ZEN file is automatically regenerated.

Object Surface #, Face # This control selects the desired surface number. Changing the selection in this control will update the NSC Object Viewer if the viewer is opened from the "View Object" button at the bottom of the tab. The screenshot below is of sample part "LB_T67C_190608_GEOMETRY.STEP" with Face 0 selected in the Object Viewer. Note more than one surface may be selected at one time. This allows groups of surfaces to be selected and renumbered together using the "Change To->" button. An alternate method to select surfaces is to click on the surface in the NSC Object Viewer.
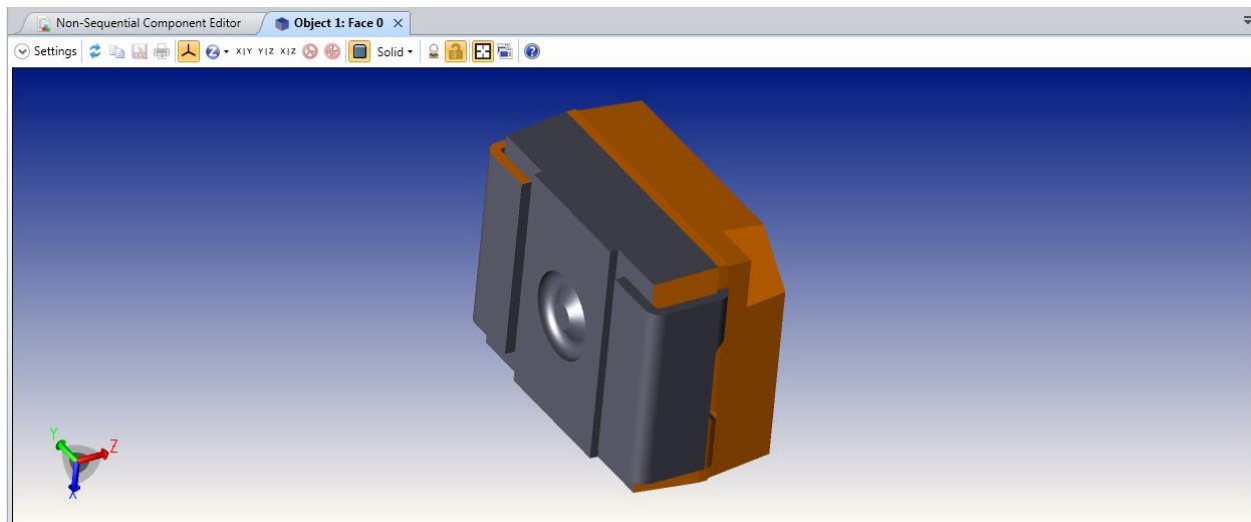
Change To-> Face # Pressing "Change" will change the highlighted surface numbers to the selected Face number.

Reset All Automatically renumbers all surfaces to the default. The default is to assign each surface a unique Face number starting at 0 through the maximum number of Faces supported. All remaining surfaces are assigned Face number 0.

Select All Selects all surfaces.

Clear All Unselects all surfaces.

Invert All Reverses the selected and unselected surfaces.

Viewer Highlights Sets the NSC Object Viewer to highlight in red all selected surfaces or all selected faces. Note if multiple surfaces are assigned the same face number, then highlighting by face will show all surfaces with the same face number.

To change how Zemax initially determines the number of unique surfaces and sets the default face numbering, see "CAD Part: STEP/IGES/SAT".

If the object is a SolidWorks part, Autodesk Inventor part, or Creo Parametric part, this is also where the associated controls are found.

Model Parameters

Chord Tolerance: This setting only affects the rendering of the object. To render the object, Zemax generates a list of triangles which approximate the object shape. The tolerance is the maximum allowed distance in lens units between a single triangle and the actual surface of the object. More triangles are added if the tolerance is set smaller which yields more accurate rendering, at the expense of speed and a larger memory requirement. The default value of zero will use a chord tolerance related to the size of the object sufficient to generate a coarse approximation of the object shape that will render quickly.

Surface Tolerance: This parameter is similar to Chord Tolerance in that it is an input used by the modeling library when trying to fit its internal representation of 3D objects and surfaces to the definition in the CAD file. However, where chord tolerance is simply the maximum allowed delta between the surface of a triangle and the curved surface, Surface Tolerance has different meanings for different types of surface representations. If Zemax fails to read a CAD model, try changing the Surface Tolerance to different integer values from 1 to 4. The default value for this input is 0, and in many cases this should be fine – control of this value is only needed for specific parts. Changing the value to 1, 2, or 3 decreases the tolerance by 1, 2, or 3 orders of magnitude, which may potentially provide more accuracy to the import.

Face Mode: The Face Mode setting determines how Zemax assigns face numbers to the various surfaces on the imported object. Some CAD programs create data files that have many more small surfaces than is useful for optical analysis. For example, a simple cylinder may be described in the CAD file by hundreds of small surfaces, while for optical analysis only two or three different optical properties are applied to the entire object. Rather than assign optical properties to each of the many surfaces, it is usually more convenient to group surfaces by assigning a single face number to all surfaces that form a continuous smooth portion of the object. The following Face Modes are supported to automate this numbering:

- Face Mode = 0: All surfaces are assigned face number 0. The entire object will have just one face.

- Face Mode = 1: All surfaces whose edges meet along a non-zero length curve, and whose normal vectors along the curve of contact are parallel within a user defined angle tolerance are assigned a common face number. The angle tolerance is defined by the Face Angle (parameter 8). This mode allows control over how finely the faces are numbered. If the Face Angle is set to a large value (such as 180) then all faces that touch will share a common number. Larger Face Angles yield fewer unique faces.

- Face Mode = 2: All surfaces are uniquely numbered. This mode yields the largest number of unique faces. Face Mode = 3: Retains the face numbers defined in the imported file. Some CAD files, such as those created by Zemax, have face numbers already defined. If Zemax recognizes the face numbers, they will be used. If Zemax does not detect the face numbers, the surfaces will be numbered as for Face Mode = 2.

- Face Mode = 4: All surfaces on each separate object defined in the CAD file are assigned a common face number. This option is useful for applying one property to all surfaces on each object when more than one object is defined with a single CAD file.

In all cases, the number of unique face numbers cannot exceed the maximum number of faces Zemax supports. Any faces which exceed the maximum limit will be renumbered to be face 0. Zemax also supports assignment of face numbers to imported object surfaces using the "Face" tab of the NSC Object Viewer, see the "CAD" section of the Object Properties. For visual confirmation of the face numbering, see "NSC Object Viewer".

Face Angle: The angle in degrees used by Face Mode = 1 described above.

SolidWorks Part

The controls in this section are used to determine which dimensions in a SolidWorks part should be displayed (exposed) to Zemax for potential modification. Only SolidWorks parts (see "CAD Part: SolidWorks") use tool.

Settings:

SolidWorks Part Dimension Name This control selects the desired dimension. Note that more than one dimension may be selected at one time. Once a group of dimensions have been selected, they can all be exposed to Zemax (for potential modification) or not exposed (and thus locked). Once a dimension is exposed to Zemax, the phrase "EXPOSED:" will appear in front of the dimension name.

Configuration This control determines which configuration of the SolidWorks part (SLDPRT) file will be read in by Zemax.

Expose Expose the selected dimensions to Zemax. Values for these dimensions will then appear in the parameter columns of the Non-Sequential Component Editor (NSCE).

Do Not Expose Do no expose the selected dimensions to Zemax. Any previously exposed dimensions will no longer be exposed.

Expose All Expose all part dimensions to Zemax.

Do Not Expose All Do not expose any part dimension to Zemax.

Clear All Clear any selections, i.e. deselect all dimensions.

Autodesk Inventor Part

The controls in this section are used to determine which dimensions in an Autodesk Inventor part should be displayed (exposed) to Zemax for potential modification. Only Inventor parts (see "CAD Part: AutoDesk Inventor") use this tool.

Settings:

Inventor Part Dimension Name This control selects the desired dimension. Note that more than one dimension may be selected at one time. Once a group of dimensions have been selected, they can all be exposed to Zemax (for potential modification) or not exposed (and thus locked). Once a dimension is exposed to Zemax, the phrase "EXPOSED:" will appear in front of the dimension name.

iPartFactory Number This control determines which iPart of an Autodesk Inventor Part (.IPT) file will be read in by Zemax.

Expose Expose the selected dimensions to Zemax. Values for these dimensions will then appear in the parameter columns of the Non-Sequential Component Editor (NSCE).

Do Not Expose Do not expose the selected dimensions to Zemax. Any previously exposed dimensions will no longer be exposed.

Expose All Expose all part dimensions to Zemax.

Do Not Expose All Do not expose any part dimension to Zemax.

Clear All Clear any selections, i.e. unselect all dimensions.

Creo Parametric Part

The controls in this section are used to determine which dimensions in a Creo Parametric part should be displayed (exposed) to Zemax for potential modification; dimensions include parameters which are used to define the part geometry analytically. Only Creo parts (see "CAD Part: Creo Parametric") use this tool.

Settings:

Creo Part Dimension Name This control selects the desired dimension. Note that more than one dimension may be selected at one time. Once a group of dimensions have been selected, they can all be exposed to Zemax (for potential modification) or not exposed (and thus locked). Once a dimension is exposed to Zemax, the phrase "EXPOSED:" will appear in front of the dimension name.

Family Table Instance This control determines which instance in the Family Table of the Creo Parametric part (.PRT) file will be read in by Zemax.

Expose Expose the selected dimensions to Zemax. Values for these dimensions will then appear in the parameter columns of the Non-Sequential Component Editor (NSCE).

Do Not Expose Do no expose the selected dimensions to Zemax. Any previously exposed dimensions will no longer be exposed.

Expose All Expose all part dimensions to Zemax.

Do Not Expose All Do not expose any part dimension to Zemax.

Clear All Clear any selections, i.e. unselect all dimensions.

Solve Types

Most data columns in the NSC Editor support solves. Solves are functions which actively adjust specific values. Solves can be specified on position, tilt, material, and other parameter values. To set a Solve Type on a cell, click on the small cell to the right of the data and select the solve type from the drop-down list. Some solves require additional parameters.
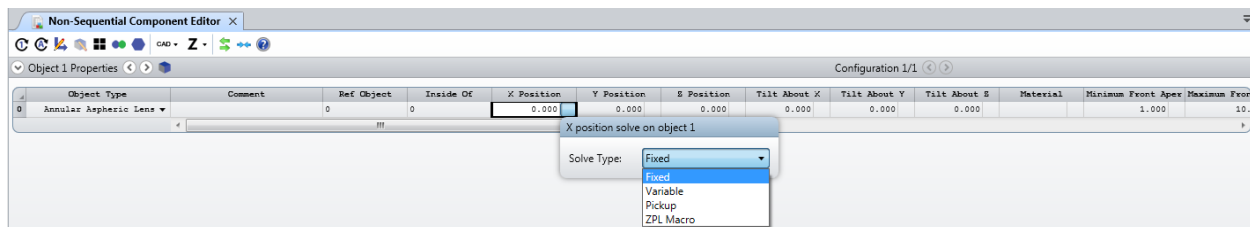
Solve Settings:

Fixed Turns solves off.

Variable To make a parameter variable for optimization, click on the small cell to the right of parameter, and select Variable in the drop down Solve Type menu. Alternatively, click on the parameter and press Ctrl-Z on the keyboard. Ctrl-Z also removes the variable status, and acts like a toggle.

Pickup The pickup solve uses a value from another object and column as the value on the target object. Pickup solves on parameters are only allowed if the source object is a preceding object, or if the source object is the same object and the source parameter column is to the left of the destination column.

ZPL Macro ZPL Macro solves call a user defined ZPL Macro to determine the solve value. Any numerical value that can be computed in a macro can be returned to the editor calling the solve macro. For information about the ZPL Macro language, see "An Overview of ZPL" and "Using ZPL Macro solves".

Merit Function Editor (from the Setup Tab)

See "Merit Function Editor" under the Automatic Optimization section of "The Optimize Tab".

Optimization Wizard (from the Setup Tab)

See "Optimization Wizard" under the "Automatic Optimization" section of "The Optimize Tab"

Non-Sequential Optimization Wizard (from the Setup Tab)

See "Non-Sequential Optimization Wizard" under the "Automatic Optimization" section of "The Optimize Tab (Non-Sequential UI Mode)"

Non-Sequential Bitmap Wizard (from the Setup Tab)

See "Bitmap Wizard" under the "Automatic Optimization" section of "The Optimize Tab (Non-Sequential UI Mode)"

Non-Sequential Roadway Lighting Wizard (from the Setup Tab)

See "Roadway Lighting Wizard" under the "Automatic Optimization" section of "The Optimize Tab (Non-Sequential UI Mode)"

Multiple Configuration Editor

The Multiple Configuration Editor button is available in the Editors section of the Setup tab, and is very similar to the Lens Data Editor.
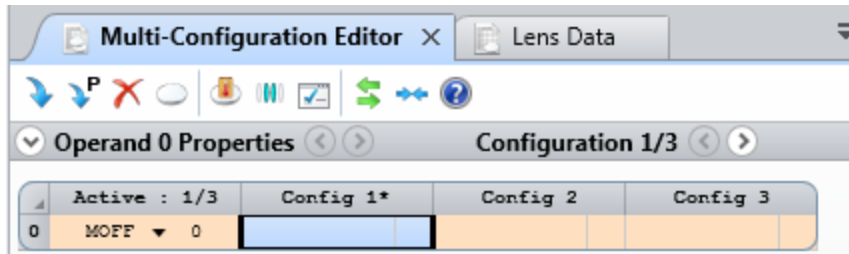
Zemax supports a very general capability for defining, analyzing, and optimizing optical systems which are used in multiple configurations. Zemax uses a substitution procedure for defining multiple configurations. The configurations are differentiated by different values for the same parameter. For example, in a zoom lens, the spacings between various elements may take on more than one value. Each set of values used together forms one configuration.

For more information see the Using Multiple Configurations help file.

In the Multiple Configuration Editor (MCE), both configurations and operands can be inserted and deleted. The insert and delete keys will also add or remove new operand rows. The data entered on the MCE will be saved automatically whenever you save your lens file.

Settings:

Operand Type Operands are listed in rows. Cells in the first column list the operand type.

The operand can be changed by clicking the drop down arrow and selecting from the list of multi-configuration operands or by typing the operand name into the cell. Operands are inserted/deleted via the right click menu, or the insert/delete keys on the keyboard.

For a complete description of multi-configuration operands, see Using Multiple Configurations. The default operand is "MOFF" which indicates an unused operand, and may be used for entering comments.

One application for the multi-configuration feature is to design systems that can be operated with or without certain elements being present. This can easily be handled by using the "Ignore This Surface" feature which uses the operand IGNR to ignore different surfaces in different configurations.

Configuration Configurations are listed in columns. Configurations are inserted/deleted via the insert/delete buttons in the toolbar.

Active Configuration The active configuration is indicated in the header of the first column. The header of the column with the active configuration also lists a "*" next to the configuration number. To set an active configuration, double click the column header at the top of the configuration you wish to select. The active configuration is also changed via the previous/next configuration arrows in the title bar, or with the shortcut keys Ctrl-A and Shift-Ctrl-A.

Other Data To edit the contents of a cell, move the cursor to that cell and type in the new data.

Operand Properties

Clicking the down arrow in the Operand Properties bar above the Multi-Configuration Editor will expand the Operand Properties dialog box. The Operand Properties shown in the dialog box can be changed by clicking on a different surface in the Multi-Configuration Editor, or by clicking the left/right arrows in the Operand Properties bar. The following properties may be defined in this dialog box.
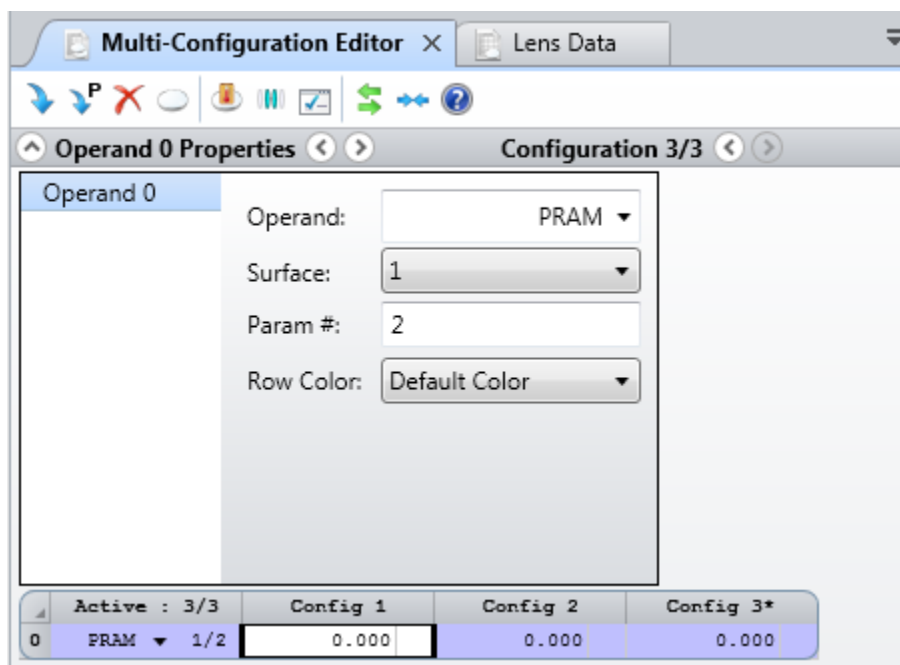
Operand Type: The operand can be changed by clicking the drop down arrow and selecting from the list of multi-configuration operands or by typing the operand name into the cell. If the operand requires any other data, this is where it is entered. See a table of multi-configuration operands under the Using Multiple Configurations help file.

Row Color: Selects either default or none for the row color for each row in the spreadsheet. Individual row colors may be set using the row color setting on the operand type dialog box.

Menu Options

The Multi-Configuration Menu Options are accessed by right clicking on the operand or cell.

Settings:

Copy Cells Copy cell data to the Windows Clipboard.

Paste Cells Paste cell from the Windows Clipboard to the current cell(s).

Cut Operands Copies all the data for a single operand or a range of operands to the Windows Clipboard, then deletes the operands. The operand or range of operands must first be selected using either of the following techniques:

Using the mouse: Click on the first operand to be selected. Hold down the left mouse button, and drag the mouse cursor to cover the range of operands desired. The selected operands will be highlighted in inverse colors. To select only one operand, drag the mouse up or down from the operand desired until two operands are selected, then move the mouse back to the desired operand.

Using the keyboard: Move the cursor to any cell on the desired operand row. Then hold down the shift key, while moving the cursor up or down until the desired range of operands is selected. The selected operands will be highlighted in inverse colors. To select only one operand, move the cursor up or down from the operand desired until two operands are selected, then move the cursor back to the desired operand.

Copy Operands Copies all the data for a single operand or a range of operands to the Windows Clipboard. To select a single operand or a range of operands, see the discussion under "Cut Operands" above.

Paste Operands Copies all the data for a single operand or a range of operands from the Windows Clipboard to the current cursor location in the Multi-Configuration Editor. The operand data must first have been copied to the Windows Clipboard using either "Cut Operands" or "Copy Operands" described above.

Insert Operand This inserts a new row in the spreadsheet at the current row. The new operand type is "OFF" which means the operand is ignored. The Insert key is a shortcut for this menu option.

Insert Operand After This inserts a new row in the spreadsheet after the current row. The new operand type is "OFF" which means the operand is ignored. The Ctrl-Insert key combination is a shortcut for this menu option.

Delete Operand This deletes the current row in the spreadsheet. The Delete key is a shortcut for this menu option.

Hide/Unhide Rows Option to hide or show the selected row(s).

Hide/Unhide Columns Option to hide or show the selected column(s).

Solve Types

To set a Solve Type on a cell, click on the small cell to the right of the data and select the solve type from the drop-down list.
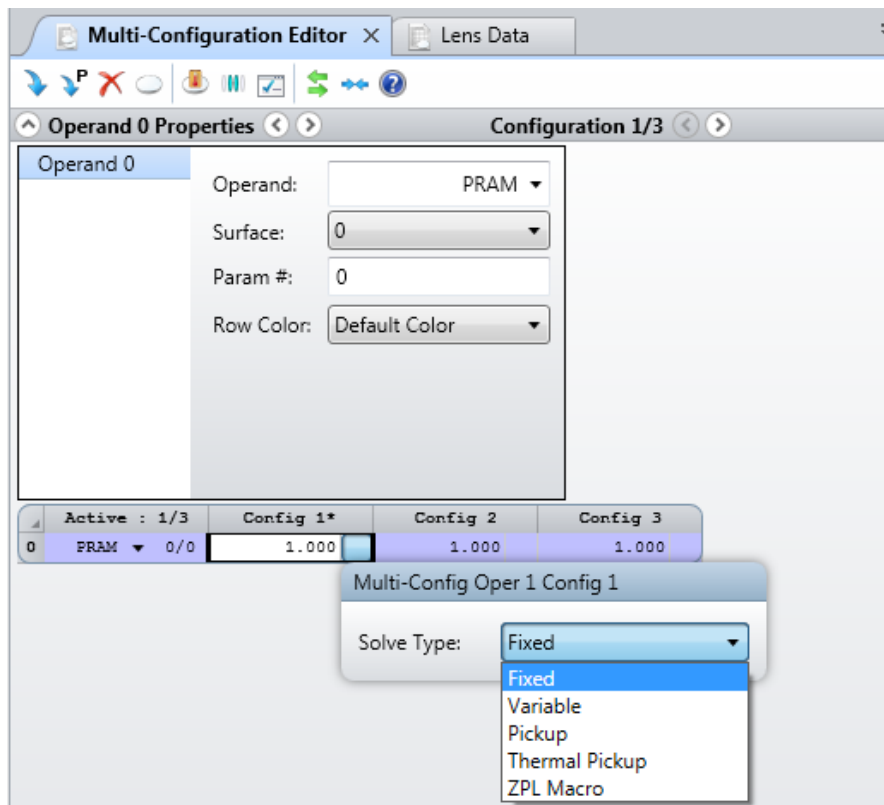
Solve Settings:

Fixed Turns solves off so the value in the MCE does not change.

Variable This applies to numerical value operands only. If the status is variable, then the optimization algorithms may adjust the value during optimization.

Substitute This applies to material operands only. If the material status is set to substitute, then the global optimization algorithms may choose another material. Optionally, a specific glass catalog to use for substitution may be specified. The name of the optional catalog is limited to 40 characters. For more information on glass substitution, see "Using glass substitution".

Pickup Suppose there are three configurations, and in two of the configurations the values for one of the rows (say a THIC or GLSS) need to be the same value. A pick-up solve on one of the configurations for that operand can be used to ensure they are always the same. To set a pickup solve, double click on the row and configuration on which the solve is to be placed. A dialog box will appear which allows definition of the solve type, the target configuration and operand, and a scale factor and offset. The new cell value is defined by: new_value = target_value*scale + offset. Note the target_value may be from any other cell in the MCE, as long as the target configuration and operand number is less than or equal to the current configuration and operand number, respectively.

Thermal Pickup This pickup solve is used to compute a new value for a multi-configuration parameter based up on the temperature and pressure of the new configuration as compared to the "reference" configuration. Thermal pickup solves only affect data for certain types of multi-configuration values, including radius of curvature and thickness (THIC). See Thermal Analysis for details.

ZPL Macro ZPL Macro solves call a user defined ZPL Macro to determine the solve value. Any numerical value that can be computed in a macro can be returned to the editor calling the solve macro. For information about the ZPL Macro language, see the An Overview of ZPL and Using ZPL Macro solves.

Multi-Configuration Editor Toolbar

The Multi-Configuration Editor Toolbar is available at the top of the Multi-Configuration Editor window. This provides quick access to tools for the Multi-Configuration Editor.

The buttons in the Multi-Configuration Editor Toolbar are listed below.

Insert Configuration

Insert a new configuration at the configuration indicated by the current position of the cursor.

Insert Configuration with Pickups

Insert a new configuration at the configuration indicated by the current position of the cursor, with pickup solves inserted to pick up values from the first configuration.

Delete Configuration

Delete the current configuration indicated by the current position of the cursor. This eliminates the column and all the data it contained.

Make Single Configuration

Delete all the multi-configuration data, leaving the lens as a single configuration system in whatever configuration is active at the time this tool is used.

Make Thermal

Set up multiple configurations of the design at different temperatures to allow thermal variation of performance to be analyzed.
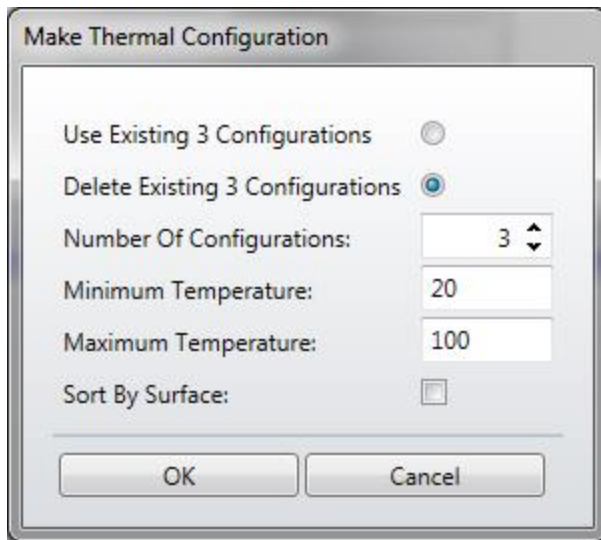
Settings:

This tool is used to perform the bulk of the tedious work in setting up a multi-configuration thermal analysis. A dialog box will appear which allows setting of the number of configurations, as well as the minimum and maximum temperatures. There are also options to delete or retain the existing multi-configuration data, and to sort the data by surfaces rather than the default sort by operand type. Be sure to read and understand how the thermal analysis works in Zemax. For more information, see "THERMAL ANALYSIS" and "Limitations of thermal analysis".

If the "Delete existing configuration data" option is selected, then 1 nominal configuration at the current temperature and pressure will be created. Additional configurations are then defined to cover the temperature range specified. If 3 configurations are requested, then there will be 1 nominal configuration (configuration 1), and 3 configurations that span the specified temperature range in equal increments, for a total of 4 configurations. The air pressure will be assumed to be the same as the nominal pressure.

If the "Use existing n configurations as nominal" option is selected, then the existing MCE data will be used to create the nominal configurations. The tool will automatically add new operands for other data typically required for thermal modeling, such as glass curvature, semi-diameters, parameter, and extra data values. In this mode, the total number of configurations will be determined by the number of nominal configurations times the number of new configurations, plus the original nominal configurations.

For each radius, thickness, glass, semi-diameter, parameter, and extra data value affected by temperature effects, appropriate operands will be inserted with TCE pickup solves. It is always good engineering practice to check the results of this automatic thermal set up tool carefully to make sure no important parameters have been omitted.

Make Conjugate

Create a subset of the optical system which models imaging between any two conjugate surfaces as a new configuration. This feature is useful for evaluating or optimizing the performance of the optical system between any two surfaces of the design.
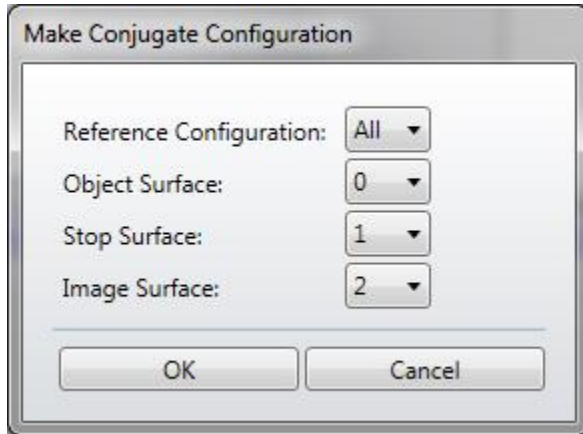
This tool will create a subset of the optical system which models imaging between any two conjugate surfaces as a new configuration. This feature is useful for evaluating or optimizing the performance of the optical system in between two surfaces which are not always the object and image surfaces. The subset system is defined by the object, stop, and image surface numbers as defined in a reference configuration. If no multi-configuration data exists, then two configurations, the nominal and subset configurations, will be created. If multi-configuration data already exists, then any configuration, or all configurations, may be used as the reference configuration(s). If a single reference configuration is selected, then one new configuration will be added. If all configurations are selected, then the total number of configurations will double, as one subset configuration will be created for each existing configuration.

The subset configurations are created by adding as required multi-configuration operands to alter the system aperture type, size, stop location, field data, thicknesses, curvatures, aspheric data, and other data as appropriate. The ignore surface feature (See "Ignore This Surface") is used to eliminate surfaces not relevant to the subset configuration. This feature makes the following assumptions about how to construct the subset system:

The system aperture is set to "float by stop size" and the semi-diameter of the new stop is set to the current semi-diameter of the surface in the reference configuration.

The field type is set to object height, and the field values are set to the primary wavelength chief ray coordinates on the specified object surface in the reference configuration.

The original object and image surfaces in the reference configuration are placed under the control of multi-configuration operands to allow those surfaces to take on the correct shape of the subset object and image surfaces. For this reason, currently the original and subset object and image surfaces must all be Standard surface types.

These assumptions may not be correct in all cases. The resulting subset conjugate system should be carefully checked for validity.

Add All Data

Add all data currently in the system, broken down by the type of data.
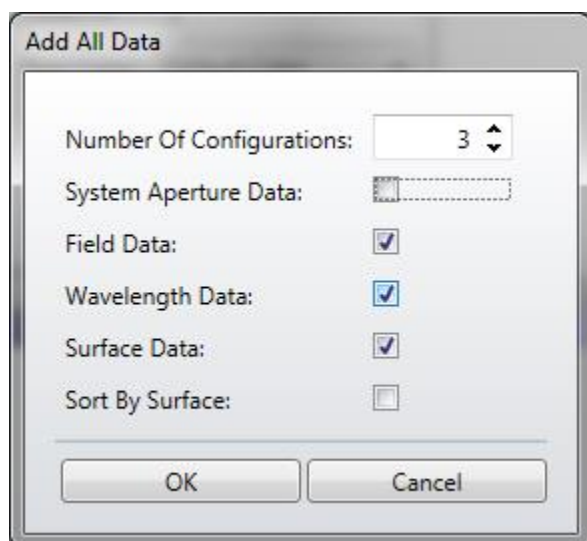
This tool will add all data currently in the system to the MCE. This eliminates the need to add many data items manually. For some systems, it is easier to delete unwanted data than to insert desired data.

Go to Operand

Go to the specified operand number.
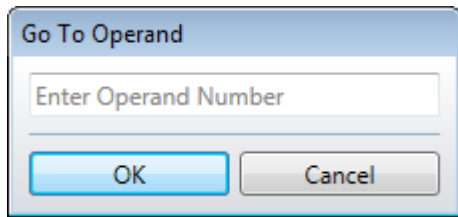
Reset Column Order

Reset column order back to default settings.

Reset Column Widths

Reset column widths to a standard size.

Help

See more help on this feature.

Tolerance Data Editor (from the Setup Tab)

The Tolerance Data editor is used to define, modify, and review the system tolerance values. See the "Tolerance Data Editor" under the "The Tolerance Tab".
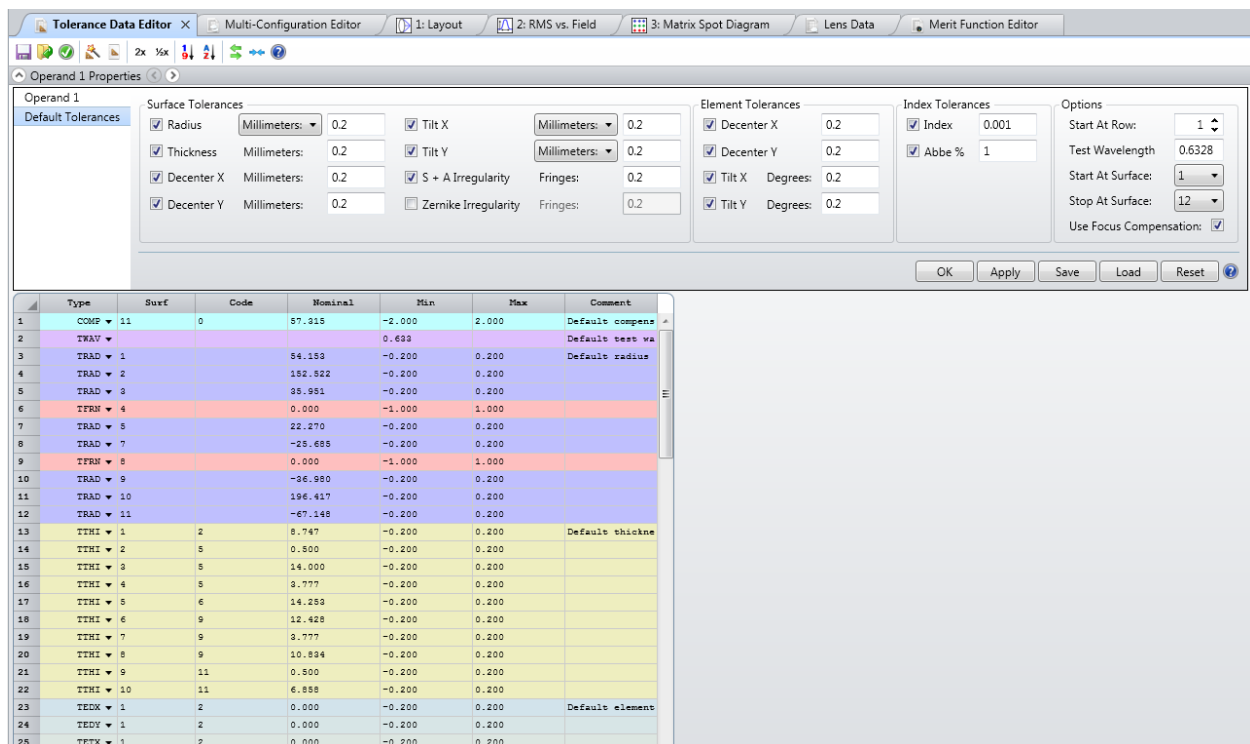
System Viewers Group

These are the System Viewers available in the Setup tab:
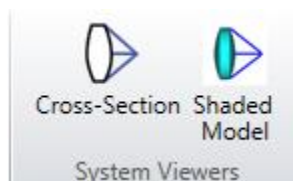
These are the System Viewers available in the Setup tab in Non-Sequential UI Mode:
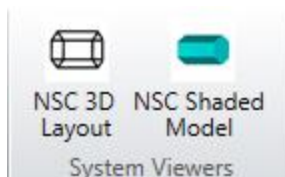
Cross-Section (from the Setup Tab)

See "Cross Section" under the "System Viewers" section of "The Analyze Tab".

Shaded Model (from the Setup Tab)

See "Shaded Model" under the "System Viewers" section of "The Analyze Tab".

NSC 3D Layout (from the Setup Tab)

See "NSC 3D Layout" under the "System Viewers" section of "The Analyze Tab (Non-Sequential Mode)".

NSC Shaded Model (from the Setup Tab)

See "NSC Shaded Model" under the "System Viewers" section of "The Analyze Tab (Non-Sequential Mode)".

Diagnostics Group

These are the Diagnostics tools available in the Setup tab:
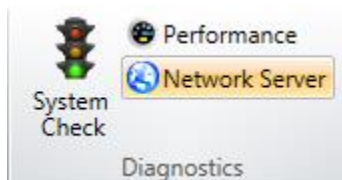
These are the Diagnostics tools available in the Setup tab in Non-Sequential UI Mode:
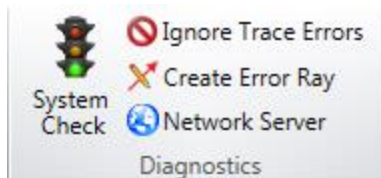
System Check

The System Check tool is available in the Diagnostics section of the Setup tab. This tool generates a data about the lens system in a separate window.

The following system check was generated with the Zemax sequential objectives sample file "Double Gauss 28 degree field.ZMX":
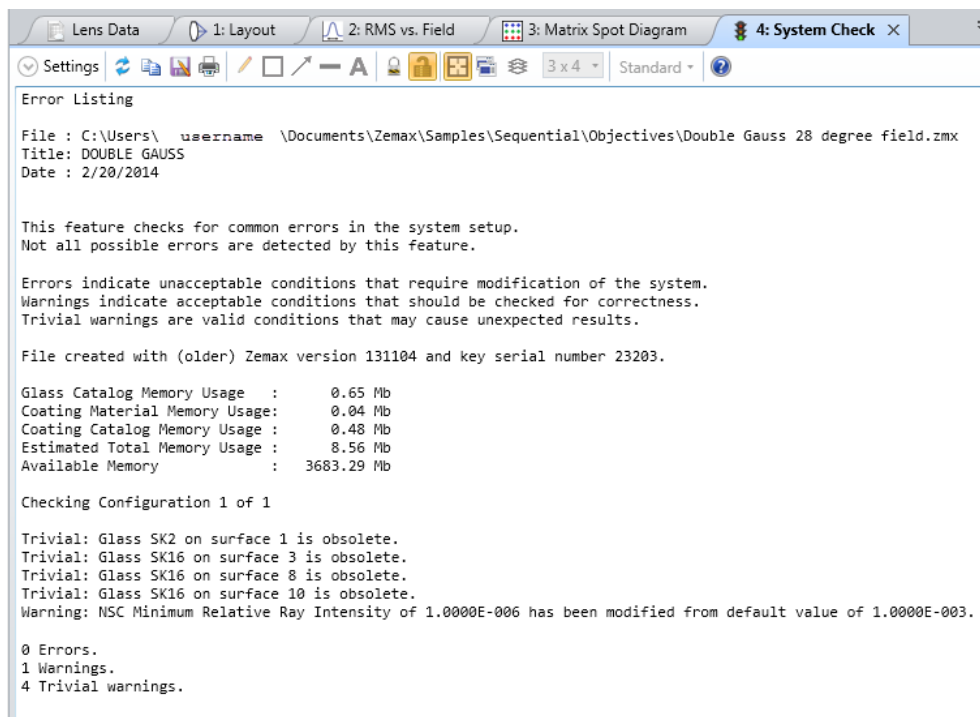
This report tests the current optical system for common errors and suspicious settings that may cause errors in analysis. The feature does not check for every possible error. Each detected potential problem is reported as either an error or a warning. Errors should be considered serious flaws in the system setup that should be corrected. Warnings are less severe, and indicate conditions that should be reviewed and the relevant settings verified for correctness.

Performance

The Performance Test tool is available in the Diagnostics section of the Setup tab. Performance test runs a check on the number of ray-surfaces per second and the number of system updates per second for the currently loaded lens file.
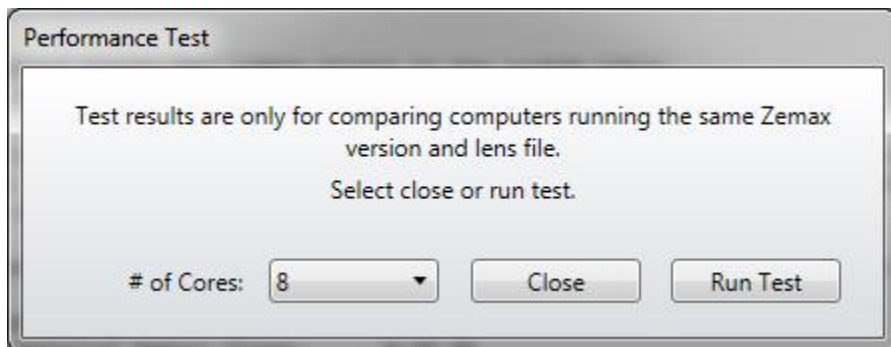
The ray surfaces per second performance number is measured by tracing a large number of random skew rays through the current optical system, and then dividing the number of rays times the number of surfaces traced through by the elapsed time in seconds.

The system updates per second is calculated by performing many system updates and then dividing the number of system updates performed by the elapsed time in seconds. System updating includes recomputing the pupil positions, field data (such as ray aiming coordinates), lens apertures, index of refraction, solves, and other fundamental checks on the lens that must be performed prior to any ray tracing.

The speed will vary tremendously depending upon the system processor, clock speed, and lens complexity.

Network Server

The Network Server test tool is available in the Diagnostics section of the Setup tab.
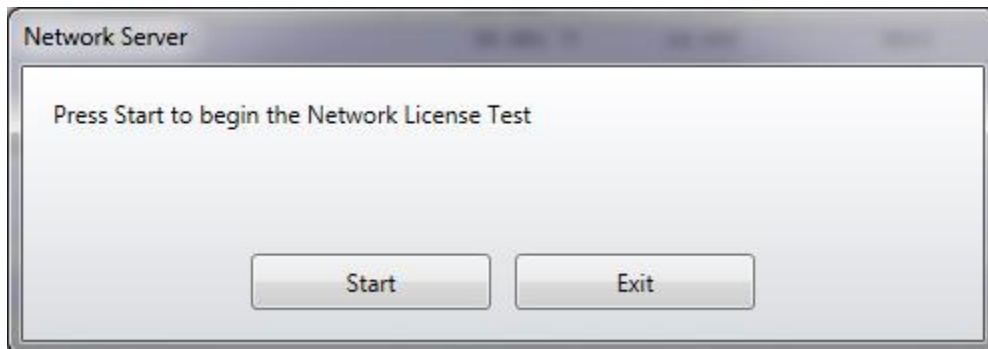
This feature is used to confirm the proper installation and configuration of the Zemax network key server. A network key is not required for this test, so it can be used prior to converting to a network license to ensure correct network operation.

NSC Ignore Trace Errors

If checked, ray trace errors that occur inside of a non-sequential group will not be reported if detected. These errors involve possibly improper placement of objects in a non-sequential group. However, in some systems it is possible to have properly defined object placement and still occasionally an error message will be generated.

Checking this option will suppress reporting of these error messages.

NSC Create Error Ray

This tool is used for isolating and studying geometry errors. For more information on geometry errors, see "Lost energy". Whenever a geometry error is displayed (not merely detected) the starting ray coordinates and cosines are stored. These starting values may be used to create a single Source Ray (see "Source Ray") that duplicates the ray with the error. If this tool is used, all sources will be set to trace zero rays for both layout and analysis. A new Source Ray will be added to the object list with the properties of the ray that generated the geometry error.

Window Control Group

These are the Window tools available in the Setup tab:

Bring to Front

The Bring to Front tool is available in the Window Control section of the Setup tab.

Selecting "Bring To Front" shows a drop down menu list of all the open windows. Any window on the list may be selected, and brought to the front of the workspace.

Window Options

The Window Options button is available in the Window Control section of the Setup tab.

These are the window display options, which set the default options for how windows are opened and positioned in the workspace.
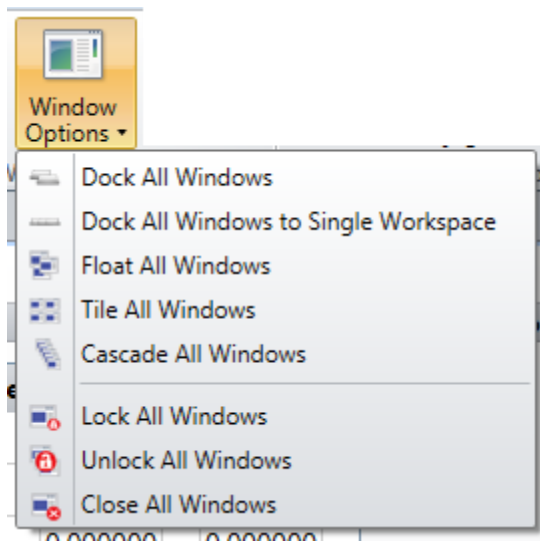
Dock All Windows Dock all windows to the workspace. Note that this may create multiple panes in the workspace.

Dock All Windows to Single Workspace Dock all windows in a single pane in the workspace.

Float all Windows Make all windows in the workspace float independently of each other.

Tile All Windows Tile all windows in the workspace.

Cascade All Windows Cascade all windows in the workspace.

Lock All Windows Prevent all open windows from being updated. Useful for generating reference data.

Unlock All Windows Unlock all open windows so they can be updated.

Close All Windows Close all open windows in the workspace.

Dock New Windows

The Dock New Windows button is available in the Window Control section of the Setup tab.

This tool toggles between docked and floating windows in the workspace.

Configuration Group

These are the Configuration tools available in the Setup tab:
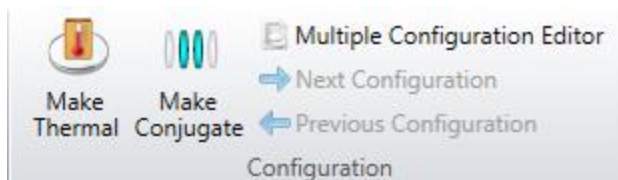
Make Thermal (from the Setup Tab)

The Make Thermal button is available in the Configuration section of the Setup tab.
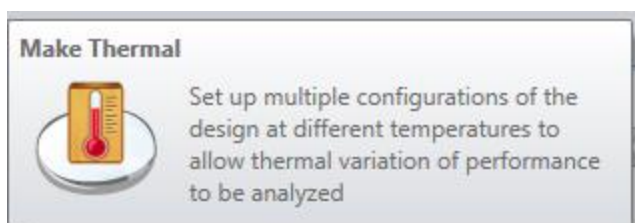
See the "Make Thermal" button in the "Multi-Configuration Editor Toolbar" under "Multiple Configuration Editor" located in the Editors section of the Setup Tab.

Make Conjugate (from the Setup Tab)

The Make Conjugate button is available in the Configuration section of the Setup tab.
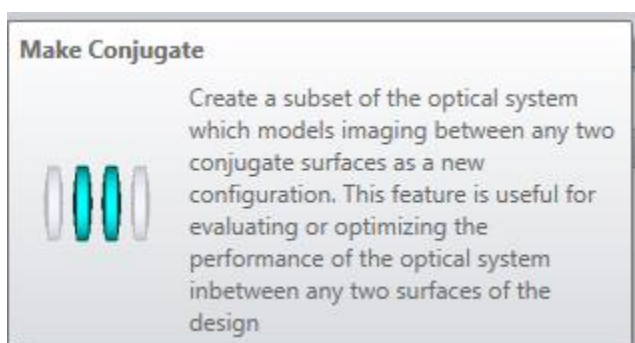
See the "Make Conjugate" button in the "Multi-Configuration Editor Toolbar" under "Multiple Configuration Editor" located in the Editors section of the Setup Tab.

Multiple Configuration Editor (from the Setup Tab)

The Multiple Configuration Editor button is available in the Configuration section of the Setup tab.
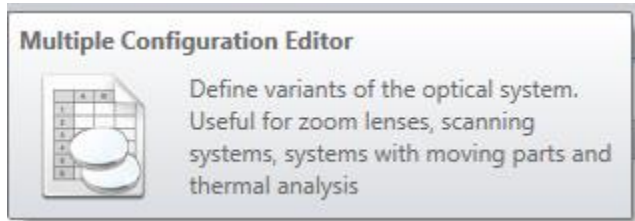
See the "Multiple Configuration Editor" located in the Editors section of the Setup Tab.

Next/Previous Configuration

The Next Configuration and Previous Configuration buttons are available in the Configuration section of the Setup tab.
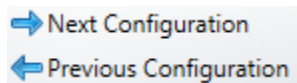
These buttons change the active configuration to the next or previous configuration defined in the Multi-Configuration Editor. See "Active Configuration" under "Multiple Configuration Editor" located in the Editors section of the Setup Tab.

The Analyze Tab

This tab provides access to all OpticStudio's Analyze capabilities
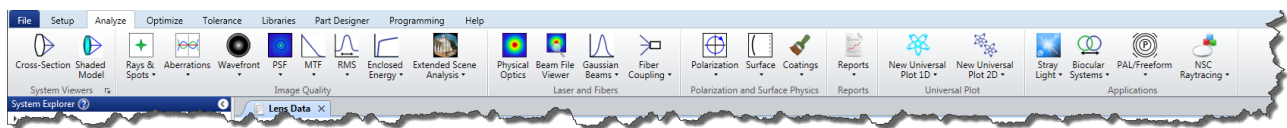
System Viewers Group
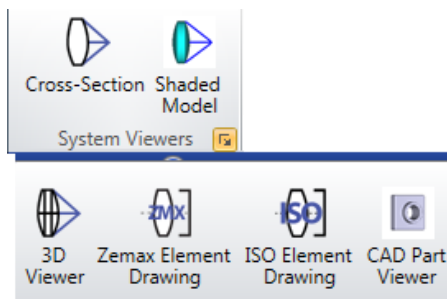
These provide the ways to view lens and elements.

Cross-Section

This is a YZ cross section through the lens.
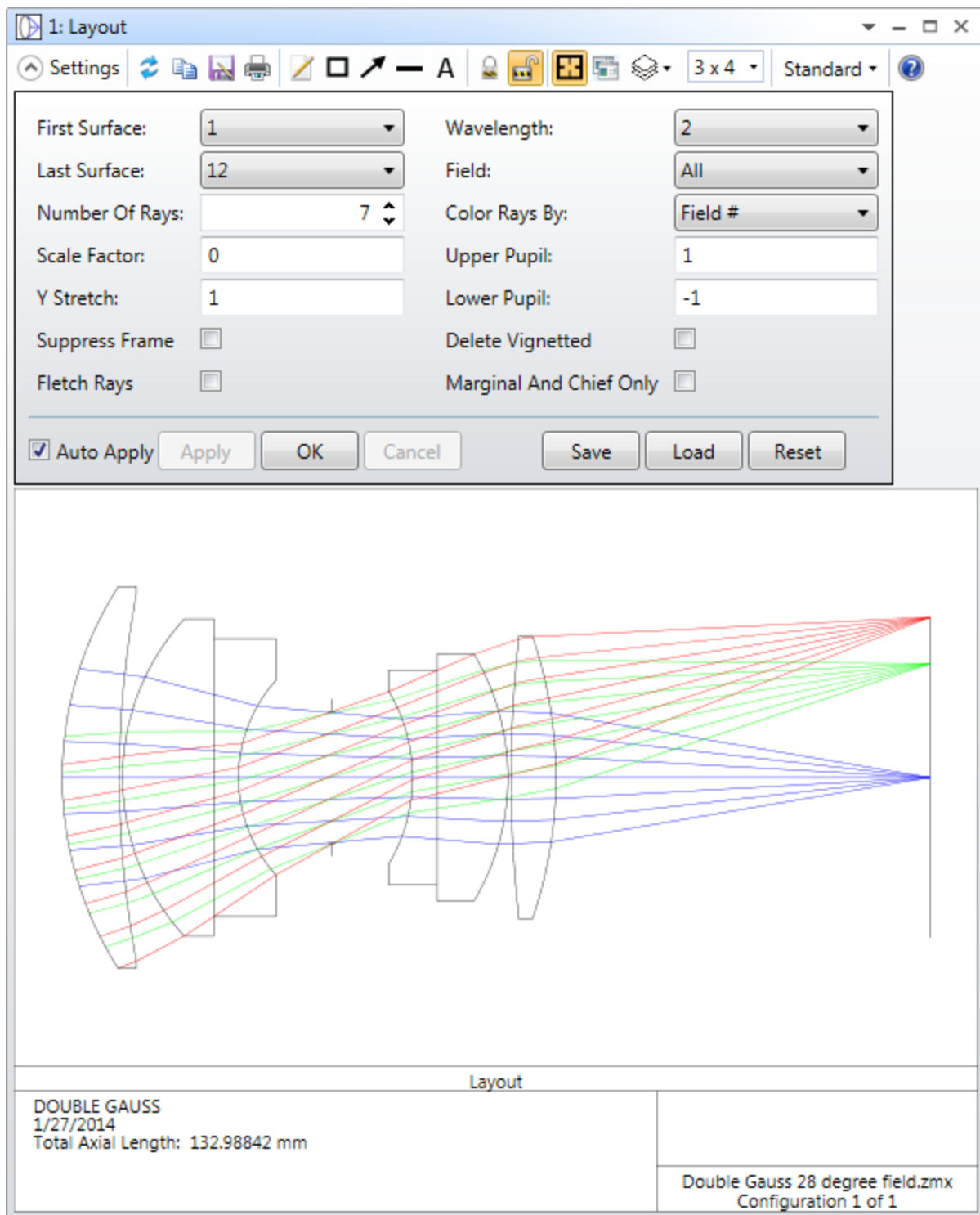
This feature is not available if you use coordinate breaks, spider obscurations, obscuration decenters, X angles, holograms, or other attributes which spoil the rotational symmetry of the lens. Use the 3D layout instead. If rays miss a surface, then the rays will not be drawn to the surface where the error occurred. If the ray is total internal reflected, then the ray will be drawn up to but not past the surface where the error occurred. Ray failures can be evaluated in detail by using "Ray Trace".

First Surface The first surface to be drawn.

Last Surface The last surface to be drawn.

Wavelength Either any one or all wavelengths may be shown.

Field Either any one or all field positions may be shown.

Number of Rays The number of rays specifies the number of tangential rays to be drawn for each defined field. The rays will be evenly distributed along the fan of the pupil, unless apodization has been specified. This parameter may be set to zero.

Scale Factor If the scale factor is set to zero, then "Fill Frame" will be selected, which will scale the range of surfaces drawn to fill the graphic page. If a numeric value is entered, then the plot will be drawn in "real" scale, times the scale factor. For example, a scale factor of 1.0 will plot the lens actual size on the printer (not the display). A factor of 0.5 will plot the lens at half scale.

Y Stretch The relative exaggeration of the Y direction. This control is useful for drawing systems with very large aspect ratios. Ignored if negative, zero, or unity.

Upper Pupil Limit The maximum pupil coordinate to draw rays to.

Lower Pupil Limit The minimum pupil coordinate to draw rays to.

Marginal and Chief Only Draws only the marginal and chief rays, overriding the other ray settings.

Color Rays By Select "Field #" to use color to distinguish between each field position, or "Wave #" to distinguish between each wavelength, "Config #" to distinguish between configurations, and "Wavelength" to approximate the color of wavelengths in the visible spectrum.

Suppress Frame Suppresses drawing of the frame on the bottom of the window, which leaves more room for the layout plot itself. No scale bar, address block, or other data will be displayed.

Delete Vignetted If checked, rays are not drawn if they will be vignetted by any surface.

Fletch Rays If checked, small arrows are drawn on each ray to indicate the direction of propagation.

Shaded Model

Draws a shaded solid representation of the lens.
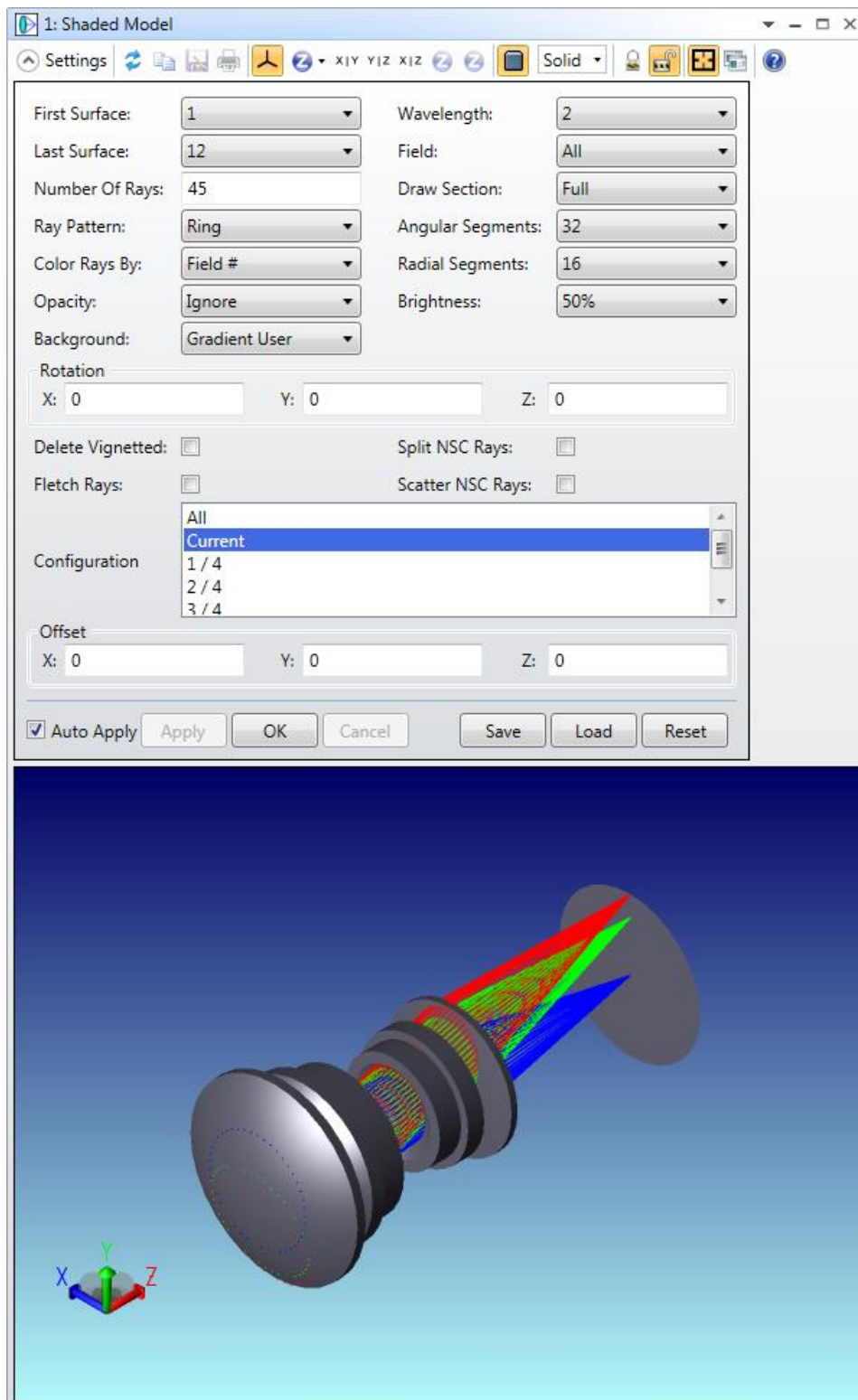
When rendering NSC detector objects with more than 1,200,000 pixels will not show individual data for each pixel to keep the total amount of required computer memory reasonable.

First Surface The first surface to be drawn.

Last Surface The last surface to be drawn.

Wavelength Either any one or all wavelengths may be shown.

Field Either any one or all field positions may be shown.

Number of Rays The number of rays specifies the number of rays to be drawn for each selected field and wavelength. The rays will be evenly distributed along the fan of the pupil, or around the perimeter if Ring is the selected Ray Pattern, or unless apodization has been specified. This parameter may be set to zero. It is ignored if the Ray Pattern is set to List.

Draw Section Select "Full" to draw each lens element completely. The 3/4, 1/2, and 1/4 options draw just that much of the element, yielding a cut away perspective of the lens interior. This only affects sequential surface data, and not NSC objects.

Ray Pattern Choose XY Fan, X Fan, Y Fan, Ring, List, Random, or Grid to indicate what the pattern of rays to be traced should be. The List option indicates that the rays to be traced are user defined and listed in a file; see the discussion below for information on the ray list format. If List is selected the Number of Rays setting is ignored.

Angular Segments The number of angular segments used to approximate the lens shapes. Larger numbers require more processing time.

Radial Segments The number of radial segments used to approximate the lens shapes. Larger numbers require more processing time.

Rotation About X The angle in degrees by which the lens appears to be rotated about the X axis.

Rotation About Y The angle in degrees by which the lens appears to be rotated about the Y axis.

Rotation About Z The angle in degrees by which the lens appears to be rotated about the Z axis.

Color Rays By For sequential rays, "Field #" will use color to distinguish between each field position, "Wave #" to distinguish between each wavelength, "Config #" to distinguish between configurations, and "Wavelength" to approximate the color of wavelengths in the visible spectrum.

For non-sequential rays "Field #" will use color to distinguish between each source number, unless a source has a user defined color assigned to that source.

Background Selects the background color.

Opacity The options are ignore, which ignores the opacity values and draws all surfaces and objects as opaque; consider, which draws partially transparent surfaces and objects based on settings in the surface and object properties settings dialog boxes; or All 50%, which overrides the defined opacity and uses 50% for all objects.

Brightness This control varies the overall brightness of the image.

Delete Vignetted If checked, rays are not drawn if they will be vignetted by any surface.

Configuration Select "All" to show all configurations, "Current" to show just the active configuration, or select any combination of other configurations.

Offset X, Y, Z The X, Y, and Z direction offset between configurations in lens units. Only has an effect on the drawing if "All" configurations are being drawn.

Fletch Rays If checked, small arrows are drawn on each ray to indicate the direction of propagation.

Split NSC Rays If checked, rays from NSC sources will be statistically split at ray-surface intercepts. Rays entering from the entry port are not affected by this setting.

Scatter NSC Rays If checked, rays from NSC sources will be statistically scattered at ray-surface intercepts. Rays entering from the entry port are not affected by this setting.

3D Viewer

Draws 3D layout plots of the lens system. The algorithm draws a wireframe style representation of the lens.
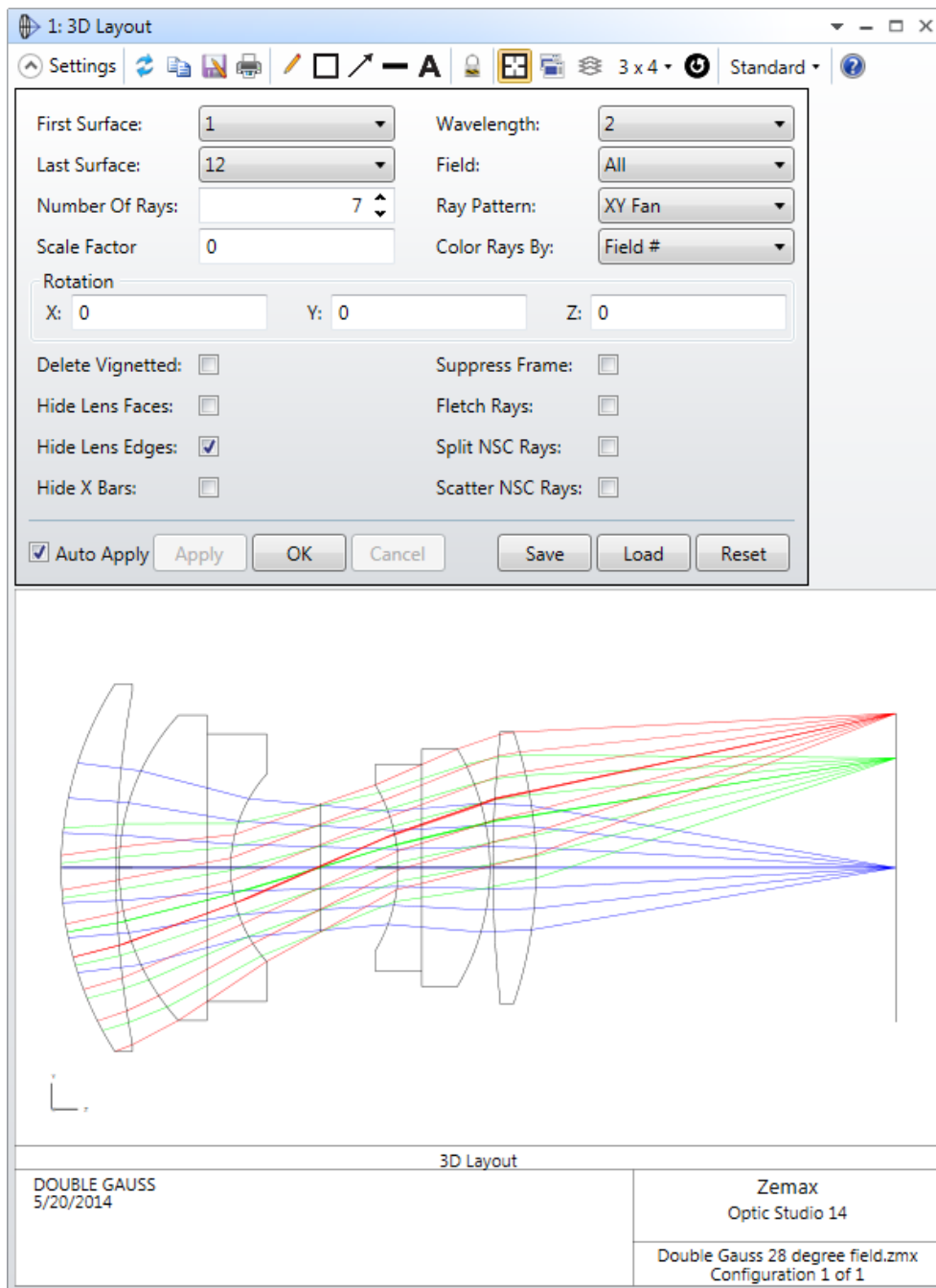
First Surface The first surface to be drawn.

Last Surface The last surface to be drawn.

Wavelength Either any one or all wavelengths may be shown.

Field Either any one or all field positions may be shown.

Number of Rays The number of rays specifies the number of rays to be drawn for each selected field and wavelength. The rays will be evenly distributed along the fan of the pupil, or around the perimeter if Ring is the selected Ray Pattern, or unless apodization has been specified. This parameter may be set to zero. It is ignored if the Ray Pattern is set to List.

Ray Pattern Choose XY Fan, X Fan, Y Fan, Ring, List, Random, or Grid to indicate what the pattern of rays to be traced should be. The List option indicates that the rays to be traced are user defined and listed in a file, see the discussion below for information on the ray list format. If List is selected the Number of Rays setting is ignored.

Scale Factor If the scale factor is set to zero, then "Fill Frame" will be selected, which will scale the range of surfaces drawn to fill the graphic page. If a numeric value is entered, then the plot will be drawn in "real" scale, times the scale factor. For example, a scale factor of 1.0 will plot the lens actual size on the printer (not the display). A factor of 0.5 will plot the lens at half scale.

Hide Lens Faces If checked, this option will suppress drawing of the lens faces, and only the lens edges will be drawn. This is useful because some complicated systems look cluttered with the faces drawn.

Hide Lens Edges If checked, this option will suppress drawing of the outer aperture of the lens. This is useful for giving the 3D layout a 2D "cross section" appearance.

Hide X Bars If checked, this option will suppress drawing of the X portions of the lens faces. This option is useful when "Hide Lens Edges" is checked and "Hide Lens Faces" is not checked.

Rotation About X The angle in degrees by which the lens appears to be rotated about the X axis.

Rotation About Y The angle in degrees by which the lens appears to be rotated about the Y axis.

Rotation About Z The angle in degrees by which the lens appears to be rotated about the Z axis.

Color Rays By For sequential rays, "Field #" will use color to distinguish between each field position, "Wave #" to distinguish between each wavelength, "Config #" to distinguish between configurations, and "Wavelength" to approximate the color of wavelengths in the visible spectrum.

For non-sequential rays "Field #" will use color to distinguish between each source number, unless a source has a user defined color assigned to that source.

Suppress Frame Suppresses drawing of the frame on the bottom of the window, which leaves more room for the layout plot itself. No scale bar, address block, or other data will be displayed.

Delete Vignetted If checked, rays are not drawn if they will be vignetted by any surface.

Configuration Select "All" to show all configurations, "Current" to show just the active configuration, or select any combination of other configurations.

Offset X, Y, Z The X, Y, and Z direction offset between configurations in lens units. Only has an effect on the drawing if "All" configurations are being drawn.

Fletch Rays If checked, small arrows are drawn on each ray to indicate the direction of propagation.

Split NSC Rays If checked, rays from NSC sources will be statistically split at ray-surface intercepts. Rays entering from the entry port are not affected by this setting.

Scatter NSC Rays If checked, rays from NSC sources will be statistically scattered at ray-surface intercepts. Rays entering from the entry port are not affected by this setting.

Discussion

Pressing the left, right, up, down, Page Up, or Page Down keys will rotate the displayed image for a different perspective.

The orientation indicator To create the layout, Zemax projects the 3D coordinates of the rotated optical model onto the 2D plane of the plot. Conceptually, the optical model is first rendered in an unrotated 3D space, with +z oriented to the right, +y up, and +x away from the viewer into the page. The coordinates and orientations of all surfaces and objects are defined by the Global Coordinate Reference Surface (set in the Surface Properties Dialogue). This 3D model is then rotated by the Rotation About X, Y, and Z values specified in the settings (see "Graphics" for information about the rotation order). The resulting 3D model is then projected onto a 2D plane by plotting the z coordinate along the horizontal direction and the y coordinate along the vertical direction; the x coordinate is ignored.

To help visually define the orientation of the model, an orientation indicator is shown in the bottom left hand corner of the drawing. This indicator consists of 3 lines extending along the directions of the +x, +y, and +z axes of the global coordinate reference surface. These 3 lines rotate as the drawing is rotated. If the coordinates of the 3D orientation

indicator lines lie in the plane of the drawing, or would extend out (toward the viewer), they are drawn in black. If the orientation indicator lines would extend in (away from the viewer) then the lines are drawn in grey.

Ray errors If rays miss a surface, then the rays will not be drawn to the surface where the error occurred. If the ray is total internal reflected, then the ray will be drawn up to but not past the surface where the error occurred. Ray failures can be evaluated in detail by using "Ray Trace". These comments only apply to rays from the sequential object surface, and not to rays from NSC sources.

Configuration data When drawing all configurations, an offset may be added to each configuration in the x, y, and z directions independently. The offsets may all be zero if desired. If the offsets are zero, then all the configurations are superimposed; otherwise, the configurations are all displaced from one another by the specified amount. Note that all offsets are defined from the global coordinate reference surface position. The global coordinate reference surface is defined in the Miscellaneous section of the System Explorer. If all offsets are zero, the multiple configurations are all overlapped at the global coordinate reference surface.

Raylist file format If List is chosen for the ray pattern, the rays to be traced are defined in a file. The file must be called RAYLIST.TXT and be placed in the <data>\Miscellaneous folder. The file format is ASCII, with two distinct methods for defining the rays supported, implicit and explicit. The implicit format file consists of two numbers on each line, one for the p_x and one for the p_y normalized pupil coordinates. The specified rays are traced at each defined field and wavelength selected.

Example: Four marginal rays are defined by:

0.0 -1.0

0.0 1.0

-1.0 0.0

1.0 0.0

The explicit format file consists of the word EXPLICIT followed by the values x, y, z, l, m, n, and wavenumber; where x, y, and z are the ray starting coordinates, l, m, and n are the direction cosines, and wavenumber is an integer indicating the wavelength to use. All coordinates are in object space. If the object thickness is infinity, then the spatial coordinates are relative to surface 1. If the object is not at infinity, then the coordinates are relative to surface 0. In both cases the ray itself is in the object space medium, prior to refraction into surface 1. If explicit format is used, then the field and wavelength settings are ignored, and only those rays listed in the file are traced.

Example: Three rays at wavelengths 1, 2, and 3 along the Y axis parallel to the Z axis are defined as follows:

EXPLICIT

0.0 -5.0 0.0 0.0 0.0 1.0 1

0.0 +0.0 0.0 0.0 0.0 1.0 2

0.0 +5.0 0.0 0.0 0.0 1.0 3

Zemax Element Drawing

This feature creates a mechanical drawing of surface, singlet, cemented doublet, or cemented triplet elements suitable for use in optical shop fabrication.

Surface The first surface of the element to be drawn.

Show As Select either "Surface", "Singlet", "Doublet", or "Triplet".

Note File Name The name of the ASCII file which contains the notes to be appended to the notes section of the element drawing. Notes should always start at number 2, since number 1 is reserved for the units specification.

Edit Note File Clicking on this button will invoke the Windows NOTEPAD.EXE editor, which can then be used to modify the selected note file.

Radius The radius tolerance value for the specified surface.

Irregularity The irregularity tolerance value for the specified surface.

Clear Aper The clear aperture diameter of the specified surface.

Thickness The center thickness tolerance of the specified surface.

Scale Factor If the scale factor is set to zero, then "Fill Frame" will be selected, which will scale the element to fill the right half of the element drawing. If a numeric value is entered, then the plot will be drawn in "real" scale, times the scale factor. For example, a scale factorof 1.0 will plot the element at actual size on the printer (not the display). A factor of 0.5 will plot the element at half scale.

Decimals The number of decimals to use in numeric values. Press the "Reset all but surface, show as, and titles" button to reformat all displayed numbers.

Drawing Title This field is for any user defined text. The default is the lens title.

Date If left blank, the default date format is used. If any text is supplied, the exact text is used.

Drawing Name All of these fields are for user defined text. Any text may be entered. No default value.

Approved

Revision

Drawn By

Project

Note Font Size Choose Standard, Medium, Small, or Fine. These are in order of decreasing font size. The Note Font Size setting only affects the size of the note file that is annotated on the drawing. Smaller fonts permit larger note files to be displayed.

Save As/Load From These two buttons allow saving and reloading of all settings displayed on the element drawing settings box. The data is stored in a binary format with the extension ELE.

Reset all but... If selected, this button will reset all the default tolerances and apertures for the specified surfaces, but the current surface, show as, and text titles will remain as they are.

Discussion The element drawing settings may be stored for the specific lens file by pressing the Save button. Unlike most analysis features, the element drawing feature saves all the settings for each surface separately. For example, the notes and tolerances for surface 1 may be saved, and then new notes and tolerances for surface 3 may be entered and then saved. To recall the settings for any specific surface, change the surface number to the desired surface, and then press the Load button. If a match is found with a previously saved surface, the settings for that surface will be displayed. This feature makes it easy to regenerate complex drawings for multiple element systems.

An important feature of the element drawing capability is the ability to load different note files and place them on the drawing. The default note file "DEFAULT.NOT" is a generic set of notes which will rarely be useful as is. However, the user can modify the note files (they are text files which any word processor or text editor can modify) and store them under different names. For example, you may want to have a .NOT file for each type of optic you design, and then load the most appropriate note file when the element drawing is generated. The .NOT files must be placed in the <data>\Miscellaneous folder (see "Folders").

If the first line of the note files starts with the number 1, then Zemax will print the note file as provided, with no default notes. If the note file starts with any other symbol, the first default note will be "1) All dimensions in millimeters" or whatever the current lens units are. To use the default note, start the user supplied notes at note number 2. The line breaks and spacings in the note file will be replicated exactly on the element drawing.

Whenever a new element drawing is generated, or the "Reset" button is pressed, the default settings will be regenerated. The default tolerances are taken from the tolerance data editor. The maximum of the min/max tolerance range is used as the default. For example, if the TTHI thickness tolerance is -.03, +.05, the tolerance value will be 0.05. Only TTHI, TRAD, and TIRR tolerances are considered. If a suitable default cannot be generated, the tolerance is set to zero. Note all tolerance fields are text; and may be edited to suit any requirement.

A handy conversion between radius tolerance and the power tolerance in fringes for a Newton's rings type optical test against a test plate is given by

$$\# \, fringes = \frac{\Delta R}{\lambda} \frac{\rho^2}{R^2}$$

where ΔR is the radius error, λ is the test wavelength, ρ is the radial aperture, and R is the radius of curvature. This formula is an approximation for shallow curvatures. For more information, see Malacara, Optical Shop Testing, J. Wiley & Sons, Inc.

Special characters There are useful special characters that may be inserted and edited in the text fields of the tolerances. The symbols are inserted by pressing and holding Alt on the keyboard and typing a 4 digit number on the numeric keypad. Some common characters are ± - Alt 0177, µ - Alt 0181, ® - Alt 0174, © - Alt 0169.

ISO Element Drawing

This feature creates an ISO 10110 type drawing of surface, singlet, or doublet elements suitable for use in optical shop fabrication.

First Surface The first surface of the element to be drawn.

Show As Choose surface, singlet, or doublet.

Text Scale This input is a scale factor for modifying the size of text displayed in the analysis; the default value is 0.7. Making the scale factor larger will increase the text size in the graphic, but making this value too large can then leave little room for the drawing of the lens itself, affecting the scale of the resulting drawing.

Other settings There are many other settings available for this feature that correspond to the tolerances defined in the ISO 10110 specification. See the discussion for details.

The ISO 10110 Element Drawing is an interpretation of the drawing specification "ISO 10110 Optics and Optical Instruments -- Preparation of drawings for optical elements and systems: A User's Guide", by Ronald K. Kimmel and Robert E. Parks, eds., published by the Optical Society of America. For more information see OSA's web site at www.osa.org.

Although the ISO specification only covers the case of single elements, the Zemax ISO drawing also supports surfaces and doublets. However, not all the ISO drawing particulars are incorporated on a drawing of a cemented doublet. If the doublet drawing results are not satisfactory, draw the doublet as two separate singlet drawings instead. This feature does not consider the "Ignore Surface" setting and will draw all requested surfaces, whether they are ignored or not.

Summary of ISO 10110 symbols and codes Zemax does not automatically include default values for all ISO tolerances; however text fields are provided for user specification of these tolerances. The text fields have limited width, and if very long strings are placed in the text fields the text may overrun the columns on the drawing. The following table summarizes the ISO 10110 symbols and codes used by Zemax.

Summary of ISO 10110 symbols and drawing codes

Symbol or Code Number	Description
R	Radius of curvature in lens units. CC indicates concave, CX indicates convex.
Ø	Diameter in lens units.
Øe	Effective diameter in lens units.
n	refractive index
v	Abbe number
(λ)	Coating specification
P; Rq; Lmin, Lmax	Indicates a polished surface on the lens drawing. The details of the polish specification are not provided on the drawing, but are specified in the surface data. Rq = maximum permissible rms surface roughness in micrometers, Lmin/Lmax = minimum and maximum length of sampling interval for rms calculation in mm.
0/A	Stress Birefringence; A = maximum optical path difference in nm/cm.
1/NxA	Bubbles & Inclusions; N = number of bubbles and/or inclusions, A = bubble grade number (size)
2/A; B	Inhomogeneity and Striae; A = homogeneity class number, B = striae class number
3/A(B/C) or 3/A(B/C) RMSx < D or 3/- RMSx < D where x is either t, i, or a	Surface Form Tolerance; A = maximum sagitta error in fringes or - if tolerance is part of radius tolerance, B = peak-to-valley irregularity in fringes or - if no tolerance is given, C = non-spherical rotationally symmetric error in fringes. If (B/C) is replaced by A(B), no tolerance is given. For RMS tolerances, D = maximum rms error in fringes, t = total rms deviation from nominal, i = rms irregularity, a = rms asymmetry remaining after asphericity is subtracted from irregularity.
4/S(L)	Centering Tolerance; S = surface tilt angle in minutes of arc, L = lateral displacement
5/NxA; CN'xA'; LN" x A"; EA'''	Surface Imperfection Method 1; N = number of imperfections, A = grade number (square root of imperfection area), C = coating imperfection designation, N' = number of coating imperfections, A' = grade number, E = edge chips designation, A''' = chip protrusion from edge, L = long scratch designation, N" = number of scratches, A" = grade number (scratch width

	(mm))
5/TV or /TV; EA" 5/RV or RV; EA"	Surface Imperfection Method 2; T = transmissive test, V = visibility class number, R = reflective test, E,A" = edge chips, as in Method 1 above.
6/Hth;L;pdg;fp;nts x np or 6/Eth;L;nts	Laser Irradiation Damage Threshold; Hth = energy density threshold, L = laser wavelength (nm), pdg = pulse duration group, fp = pulse repetition rate (Hz), nts = number of test sites, np = number of pulses per test site, Eth = power density threshold in watt-cm^2.

Diameters and bevels The semi-diameter value is used to determine the drawn radial aperture of each surface. A flat bevel is added to any surface whose semi-diameter is less than the largest semi-diameter being drawn. To place flat bevels on all surfaces, use the "Dia (flat) optional" setting available on the "L Surf Codes 3-4" tab. This optional value must exceed twice the semi-diameter of every surface drawn, otherwise the value is ignored.

Tolerance data When the "Reset from TDE" button is pressed, radius, thickness, index, surface tilt, and surface decenter tolerances are taken from the tolerance data editor. The tolerances are automatically reset after changing the surface number or show as settings. Only TRAD, TCUR, TFRN, TCON, TTHI, TIND, TABB, TSTX, TSTY, TSDX, TSDY, TIRR, TEZI, and TWAV tolerances are considered. If a tolerance is not provided, the tolerance is set to zero. The TFRN tolerance is applied to the surface form tolerance code 3/A. Note all tolerance fields may be edited to suit any requirement.

CAD Part Viewer

This feature is only available in the Premium edition of Zemax.

Draws a shaded solid model representation of a part or assembly as defined in a CAD format file

File Name   Selects the CAD part file to draw.

Brightness   This control varies the brightness of the image.

Opacity The options are to ignore, which draws all surfaces and objects as opaque, or all 50%, which draws partially transparent objects.

Background   Selects the background color.

Rotation About X   The angle in degrees by which the lens appears to be rotated about the X axis.

Rotation About Y   The angle in degrees by which the lens appears to be rotated about the Y axis.

Rotation About Z   The angle in degrees by which the lens appears to be rotated about the Z axis.

Discussion

The object may be rotated with the mouse or the cursor keys to change the viewing angle. When selecting a SolidWorks (*.sldprt, *.sldasm), Autodesk Inventor (*.ipt, *.iam), or Creo Parametric (*.prt, *.asm) file, the file is automatically converted to either a STEP or an IGES file for viewing.

Image Quality Group
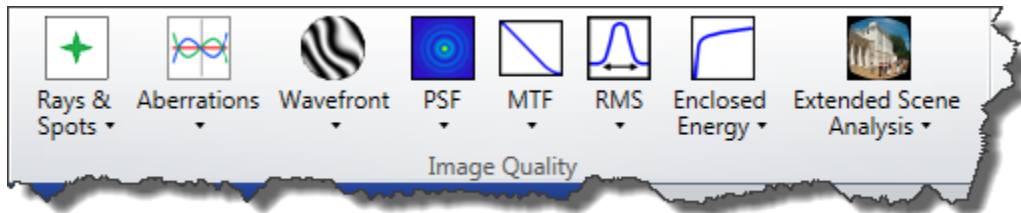
These provide ways to analyze performance and image quality of lens designs.

Rays and Spots
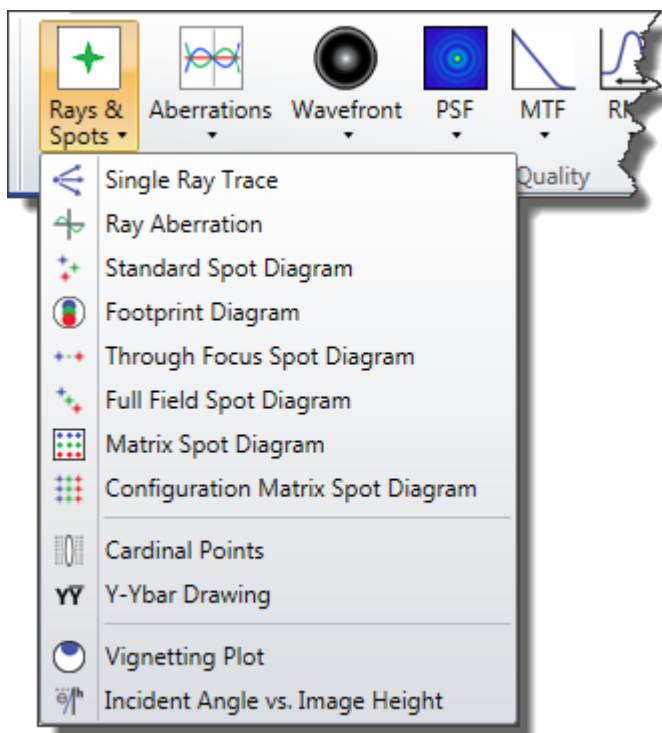
Group of analyses using geometric ray tracing of individual rays, fans of rays and bundles of rays.

Single Ray Trace

Paraxial and real trace of a single ray.

Hx Normalized x-field coordinate. The value should be between -1 and 1.

Hy Normalized y-field coordinate. The value should be between -1 and 1.

Field Select either a specific field or "Arbitrary" to enter in Hx and Hy. If a specific field is selected; the Hx and Hy controls are grayed out.

Wavelength The wavelength number of the ray to trace.

Px Normalized x-pupil coordinate. The value should be between -1 and 1.

Py Normalized y-pupil coordinate. The value should be between -1 and 1.

Global Coordinates If checked, all ray trace data is given in global coordinates rather than local coordinates, except tangent angles.

Type Select "Dir Cosines" to display the ray direction cosines at each surface, "Tangent Ang" to display the tangent of the angle the ray makes at each surface, or "Ym,Um,Yc,Uc" to display the paraxial marginal and chief ray intercept/tangent angle display. The tangent angle is the ratio of the x (or y) direction cosine to the z direction cosine.

Discussion: The Hx, Hy, Px, Py, and Global Coordinates settings are ignored if the "Ym, Um, Yc, Uc" option is selected. For a description of normalized coordinates, see the chapter "Conventions and Definitions".

For the other options, this feature allows the user to specify the normalized object coordinates, the normalized pupil coordinates, and wavelength number and then view the real and paraxial ray coordinates at every surface.

The first set of data presented is for a real ray. The values presented are the coordinates (in the surface local or the global coordinate system) of the ray intercept point. The direction cosines (or tangent angles) are the values of the ray in the surface after refraction. The direction cosine value is the cosine of the angle the ray makes with respect to the specified axis (the x-direction cosine is the cosine of the angle the ray makes with respect to the x-axis, etc.) The second set of data is just like the first set, except it is for a paraxial ray. Tangent angles are always computed relative to the local Z axis; regardless of the Global Coordinates setting.

Ray Aberration

Shows ray aberrations as a function of pupil coordinate.

Plot Scale Sets the maximum vertical scale for the plots. The maximum scale is in micrometers for ray fans, waves for OPD plots, or percent for entrance pupil aberration plots. This overrides the automatic selection of scale for the plots. Enter zero for automatic scaling.

Number of Rays This is the number of rays traced on each side of the origin of the plot.

Wavelength The wavelength number for which the calculation should be performed.

Field The field number for which the calculation should be performed.

Tangential Selects which aberration component to plot for the tangential fan. Since tangential fans are functions of the y pupil coordinate, the default is to plot the y component of the aberration.

Sagittal Selects which aberration component to plot for the sagittal fan. Since sagittal fans are functions of the x pupil coordinate, the default is to plot the x component of the aberration.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves.

Check Apertures Specifies whether or not to check if rays pass all surface apertures. If this is selected, rays which do not pass surface apertures will not be drawn.

Vignetted Pupil If checked, the pupil axis will be scaled to the unvignetted pupil, in which case the data will reflect the vignetting in the system. If unchecked, the pupil axis will be scaled to fit the vignetted pupil.

Surface Selects the surface at which the fan is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces" below.

Discussion

The tangential fans show either the x or the y component of the transverse ray aberration as a function of the y pupil coordinate of the ray. The default option is to plot the y component of the aberration. However, since transverse ray aberrations are vectors, this is an incomplete description of the aberration. When Zemax plots the y component, the plot is labeled EY, when plotting the x component of the aberration, the plot is labeled EX.

The vertical axis scale is given at the bottom of the graph. The data being plotted is the difference between the ray intercept coordinate and the chief ray intercept coordinate. The tangential fan is the plot of the difference between the ray x or y coordinate and the chief ray x or y coordinate at the primary wavelength, as a function of the y pupil coordinate. The sagittal plot is the difference between the ray x or y coordinate and the chief ray x or y coordinate as a function of the x pupil coordinate. The horizontal scale for each graph is the normalized entrance pupil coordinate, either PX or PY.

If "All" wavelengths are shown, then the plot is referenced to the primary wavelength chief ray. If monochromatic, then the chief ray for the selected wavelength is used as a reference. For this reason, the data for non- primary wavelengths will in general change when switching between monochromatic and polychromatic displays.

Because ray aberrations are vectors, with both an x and a y component, the ray aberration fan is an incomplete description of the aberrations, especially when the image surface is rotated or the system is otherwise non- rotationally symmetric. Also, the fans only indicate aberrations along two "slices" through the pupil, rather than over the entire entrance pupil. The primary purpose of the ray fan plot is to determine what aberrations are present in the system; it is not a complete description of the system performance, especially for systems without rotational symmetry.

Evaluating results at intermediate surfaces

Zemax can compute analysis results for surfaces other than the image surface by applying assumptions. These assumptions work in most cases, however, there may be cases where the methods described here for analyzing intermediate surfaces are not appropriate, including systems which require ray aiming. All of the changes made to the lens described are made on a copy of the original lens data made for analysis purposes so no changes are made to the original lens data.

If the field type is either real or paraxial image height, the field type is changed to angle or object height for infinite or finite conjugate systems, respectively. The angles and heights used correspond to the primary wavelength chief ray angles and heights as computed for the unaltered system.

If the selected surface precedes the stop surface, Zemax moves the stop surface to a (possibly virtual) dummy space prior to the existing surface 1. Unless the system aperture is object space numerical aperture or cone angle, the system aperture is changed to entrance pupil diameter, and the aperture value is set equal to the original paraxial entrance pupil diameter computed for the original stop position. Note this assumption might not be valid for systems that require ray aiming. If the selected surface follows the stop surface, no changes are made to the system aperture or stop definitions.

Surfaces which follow the selected surface are then deleted. The glass of the new image surface is set to be the same as the selected surface.

Most analysis features that compute results for focal mode systems make more sense if the rays are allowed to come to a focus after refraction from the desired surface. For example, the OPD plot is generally a useful diagnostic only when the OPD is measured at the (possibly virtual) focus of that surface. Other features which require a temporary image to be formed include PSF, MTF, and diffraction encircled energy. For these features requiring a temporary image, the new image surface is set to be a standard plane surface. A paraxial marginal ray height solve is placed on the selected surface thickness to place the image surface at paraxial focus for the selected surface. The analysis computation then proceeds at this newly created intermediate image surface. Note the analysis occurs at the paraxial focus formed by the rays after refracting through the selected surface. This shift to paraxial focus is not performed if the intermediate system is in afocal mode. For more information on afocal mode see "Afocal Image Space" in the Aperture section of the System Explorer.

Some analysis features make more sense if the data is evaluated directly on the surface itself, without allowing the rays to focus to an image. These features include the various spot diagrams, footprint analysis, geometric encircled energy, line/edge spread function, and extended source encircled energy.

Standard Spot Diagram

Traces bundles of rays through the system to the specific surface to show ray distributions.
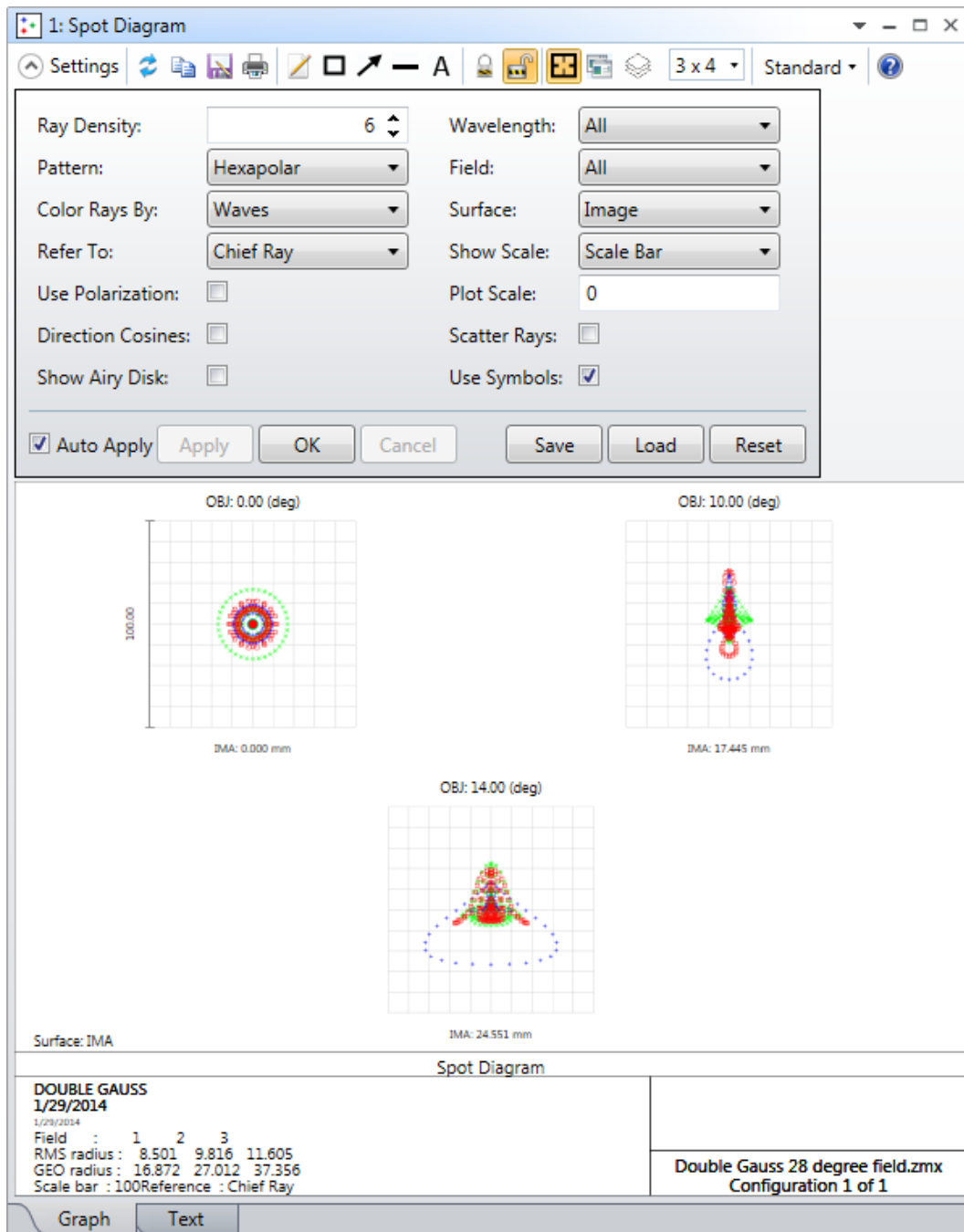
Pattern The pattern may be either hexapolar, square, or dithered. These terms refer to the pattern of rays as they appear in a pupil plane. Defocus the lens significantly to see the pattern, if desired. Dithered spot diagrams are generated by pseudo-random rays which eliminate the symmetrical artifacts in the spot diagram typical of rectangular or hexapolar patterns. The pattern is distorted to give the correct distribution of rays if pupil apodization is specified. There is no "best" pattern to use; each shows a different character of the spot diagram.

Refer To The spot diagrams by default are referenced to the real chief ray. The RMS and GEO spot radii listed at the bottom of the diagram (and defined in the discussion section) are calculated assuming the chief ray is the "zero aberration" point. However, this option allows selection of other reference points. The centroid is defined by the average position of the rays traced. The middle is defined so that the maximum ray errors are equal in the plus and minus x and y

directions. The vertex is defined by the local coordinates 0,0 on the selected surface. If the system is in afocal mode, the chief ray reference will be used instead of the vertex reference.

Show Scale Scale bar is the default. Selecting "Square" will draw a box, centered on the reference point, whose width is twice the distance from the reference point to the outermost ray. Selecting "Cross" will draw a cross through the reference point location.

The "Circle" setting will draw a circle centered on the reference point.

Wavelength The wavelength number for which the calculation should be performed.

Field The field number for which the calculation should be performed.

Surface Selects the surface at which the spot diagram is to be evaluated. This is useful for evaluating intermediate images or vignetting.

Plot Scale Sets the maximum scale size of the display in micrometers. A setting of zero will generate a default scale.

Delta Focus The delta focus option is only used if through-focus spot diagrams are selected.

For focal systems, delta focus is the Z-axis spacing between the spot diagrams in micrometers. For afocal systems, delta focus is the optical power spacing between the spot diagrams in diopters.

Five spot diagrams will be shown for each field position. The defocus will be -2, -1, 0, 1, and 2 times the delta focus provided.

Ray Density The ray density specifies the number of hexapolar rings to be traced if a hexapolar or dithered pattern is selected, or the number of rays across the width and height if a rectangular pattern is selected. The more rays traced, the greater the accuracy of the RMS spot radius, although the computation time increases. There are 6 rays in the first hexapolar ring, 12 in the second, 18 in the third, and so on.

Use Symbols If checked, this option will draw different symbols rather than dots for each wavelength. This helps distinguish the various wavelengths.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Scatter Rays If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. See "Surface scattering settings".

Airy Disk If checked, an elliptical ring around each spot showing the size of the Airy ellipse will be drawn. The Airy disk radius is 1.22 times the wavelength (primary wavelength is used if polychromatic) multiplied by the F/# of the system; which in general depends upon field position, pupil orientation, and possibly vignetting of rays if the marginal rays cannot be traced.

Direction Cosines If checked, the data presented will be the direction cosines of the rays rather than the spatial coordinates of the rays. The x direction data will be the x direction cosine of the ray, the y direction data will be the y direction cosine. The image coordinates will also be given as the reference point direction cosines. Direction cosines are dimensionless.

Configuration Select "All" to show data for all configurations at once, or select the one configuration to show, or select "Current" to show the active configuration. If "All" is selected, and both focal and afocal mode configurations are defined, the Configuration setting will automatically be reset to "Current".

Color Rays By Select "Field #" to use color to distinguish between each field position, or "Wave #" to distinguish between each wavelength, or "Config #" to distinguish between configurations, and "Wavelength" to approximate the color of wavelengths in the visible spectrum.

Discussion The ray density has a maximum value based upon the number of fields displayed, the number of wavelengths defined, and available memory. Through-focus spot diagrams will trace half of the maximum number of rays possible on standard spot diagrams.

The GEO spot radius listed on the plot for each field point is the distance from the reference point (which is either the chief ray at the primary wavelength, the centroid of all the rays traced, or the middle of the spot cluster) to the ray which

is farthest away from the reference point. In other words, the GEO spot radius is the radius of the circle centered at the reference point which encloses all the rays.

The RMS spot radius is the root-mean-square radial size. The distance between each ray and the reference point is squared, and averaged over all the rays, and then the square root is taken. The RMS spot radius gives a rough idea of the spread of the rays, since it depends upon every ray. The GEO spot radius only gives information about the one ray which is farthest from the reference point.

For information on the X and Y RMS spot radii, see the "text" listing for the spot diagram.

The Airy disk radius is given by 1.22 times the wavelength (primary wavelength is used if polychromatic) times the F/# of the beam, which in general depends upon field position and pupil orientation. This is the radius to the first dark ring of the Airy disk for a circular, uniformly illuminated entrance pupil. The Airy disk may be optionally drawn to give an idea of the scale of the plot. For example, if all the rays are well within the Airy disk, then the system is often said to be "diffraction limited". If the RMS spot radius is significantly larger than the Airy disk radius, then the system is not diffraction limited. The threshold for diffraction limited performance depends upon which criterion is used. There is no absolute boundary at which the system becomes diffraction limited. The Airy disk shown is not an accurate representation of the diffraction dark ring shape or size if the system does not have uniform illumination or if vignetting is used to eliminate some of the rays. Zemax does not plot vignetted rays on spot diagrams, nor are they used in computing the RMS or GEO spot radii.

Zemax generates grids of rays based upon the wavelength weighting factors and the pupil apodization, if any. The wavelength with the largest weight uses the maximum grid size set by the "Ray Density" option. Wavelengths with lower weights use grids with fewer rays to maintain the correct representation in the diagram. Ray grids are also distorted to maintain the correct ray distribution, if apodization is specified. The RMS spot radius stated on the spot diagram considers the wavelength weighting and apodization factors. However, it is only an estimate of the RMS spot radius based on the rays actually traced. It is not a very accurate estimate for some systems.

The image surface intercept coordinates of the reference point are shown underneath each spot diagram. If a surface other than the image surface is specified, then the coordinates are the intercept coordinates of the reference point on that surface. Since the reference point may be selected to be the centroid, this provides a convenient way of determining the centroid coordinates. If the system is using afocal mode, the image coordinates are given in radians with respect to the local Z axis.

For graphic windows, the left/right cursor keys will change the surface number and recompute the data.

Footprint Diagram

Displays the footprint of the beam superimposed on any surface. Used for showing the effects of vignetting and for checking surface apertures.
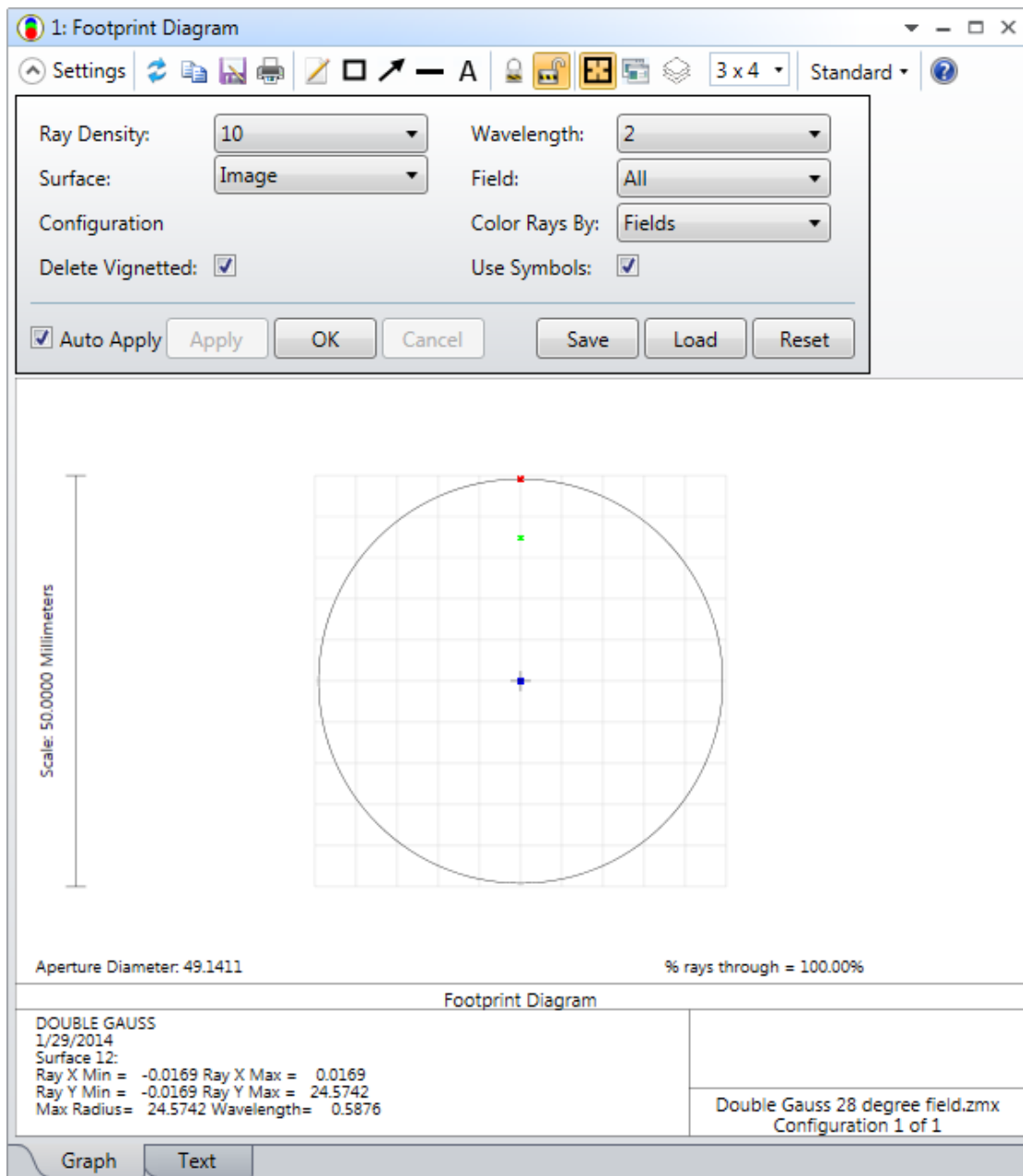
Ray Density Determines the number of rays traced across the half pupil; a setting of 10 will trace a grid of 21 x 21 rays.

The "Ring" option will trace 360 marginal rays around the edge of the pupil at each field and wavelength. The ring option will automatically determine the radial coordinate of the marginal ray that is not vignetted to approximate the shape of the beam on any surface, but will yield incorrect results if the beam is in a caustic.

Surface The surface to show the beam footprint on.

Wavelength The wavelength number to be used in the calculation.

Field The field number to be used in the calculation.

Delete Vignetted If checked, then rays which are vignetted by subsequent surfaces will not be drawn. Rays which are vignetted by prior surfaces are never drawn.

Use Symbols If checked, this option will draw different symbols rather than dots for each wavelength. This helps distinguish the various wavelengths.

Configuration Select "All" to show all configurations, "Current" to show just the active configuration, or select any combination of other configurations.

Color Rays By Select "Field #" to use color to distinguish between each field position, or "Wave #" to distinguish between each wavelength, "Config #" to distinguish between configurations, and "Wavelength" to approximate the color of wavelengths in the visible spectrum.

Discussion This feature will draw the shape of the surface, and then superimpose on that surface a grid of rays. If there is no aperture on the surface, then a circular surface shape with a radial clear aperture of the semi-diameter value is shown. Otherwise, the shape of the aperture is shown. The surface aperture is always shown as centered in the frame; even if the aperture is decentered on the actual surface. If there is an obscuration on the surface, then the obscuration will be drawn along with the circular aperture defined by the semi-diameter. If the surface aperture on the selected surface is modified between different configurations, and more than one configuration is selected, the aperture for the first configuration only will be drawn.

The ray grid size is specified by the ray density parameter, and rays may be from any or all fields, at any or all wavelengths. Rays which are vignetted by surfaces prior to the surface shown are not drawn. Rays which are vignetted by the surface or subsequent surfaces are not drawn if "delete vignetted" is checked, otherwise, they are drawn. The ray set is apodized if any system pupil apodization is selected. The number of rays shown divided by the total number of rays launched is listed as the percentage of rays through if the Ray Density is not "ring".

For graphic windows, the left/right cursor keys will change the surface number and recompute the data.

Through Focus Spot Diagram

Show spot diagrams as they change through focal plane shifts.
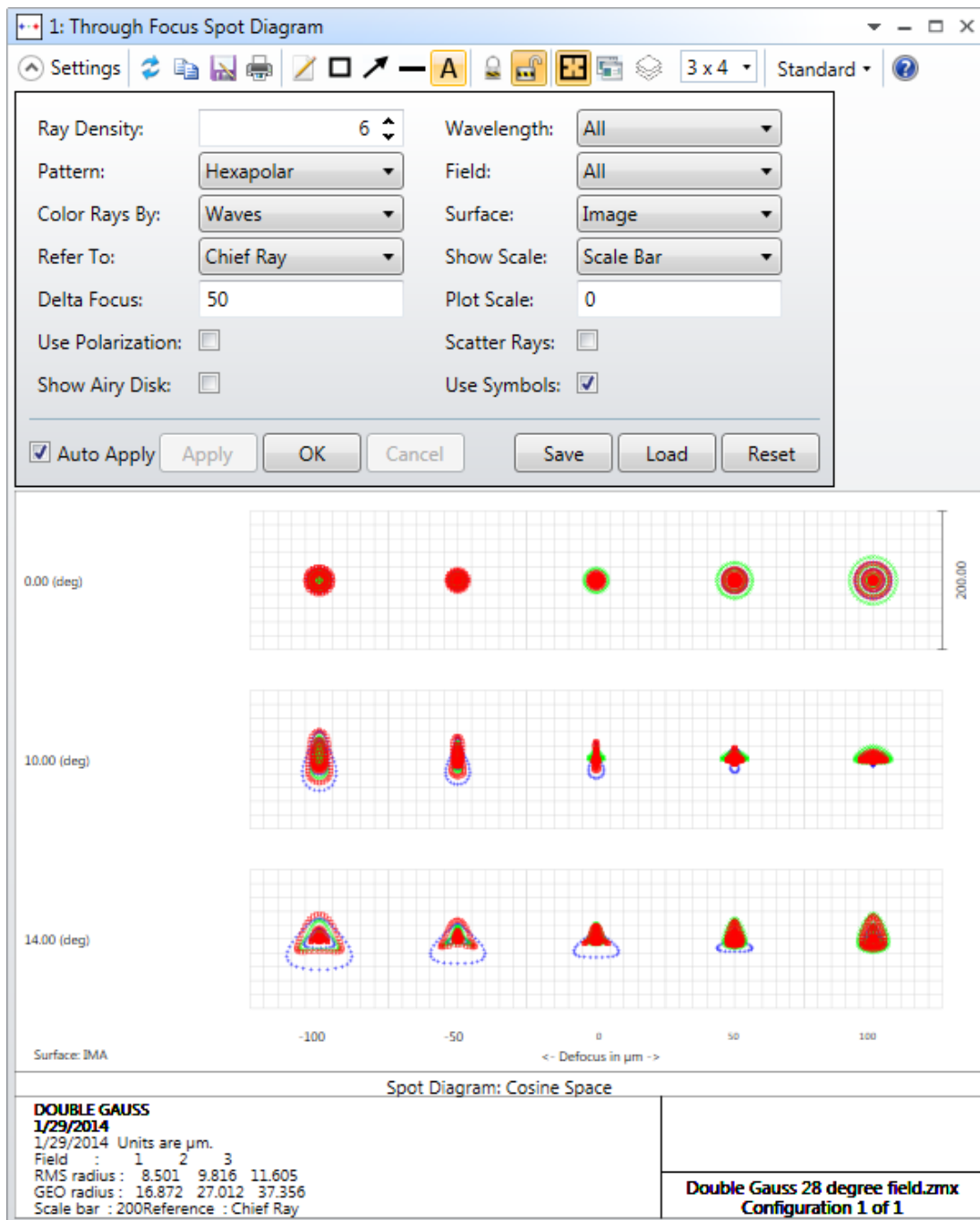
Most of the options are identical to the Standard Spot diagram.

Delta Focus: The delta focus option is only used if through-focus spot diagrams are selected. For focal systems, delta focus is the Z-axis spacing between the spot diagrams in micrometers. For afocal systems, delta focus is the optical power spacing between the spot diagrams in diopters. Five spot diagrams will be shown for each field position. The defocus will be -2, -1, 0, 1, and 2 times the delta focus provided. The delta focus units are in microns. For some systems the default delta focus may be too small to cause a change in the spot diagram structure.

Discussion The through focus spot diagrams are useful for estimating astigmatism, or for analyzing best focus or depth of focus. The Direction Cosine option is not supported.

Full Field Spot Diagram

Shows spot diagram with all field points on a common scale.
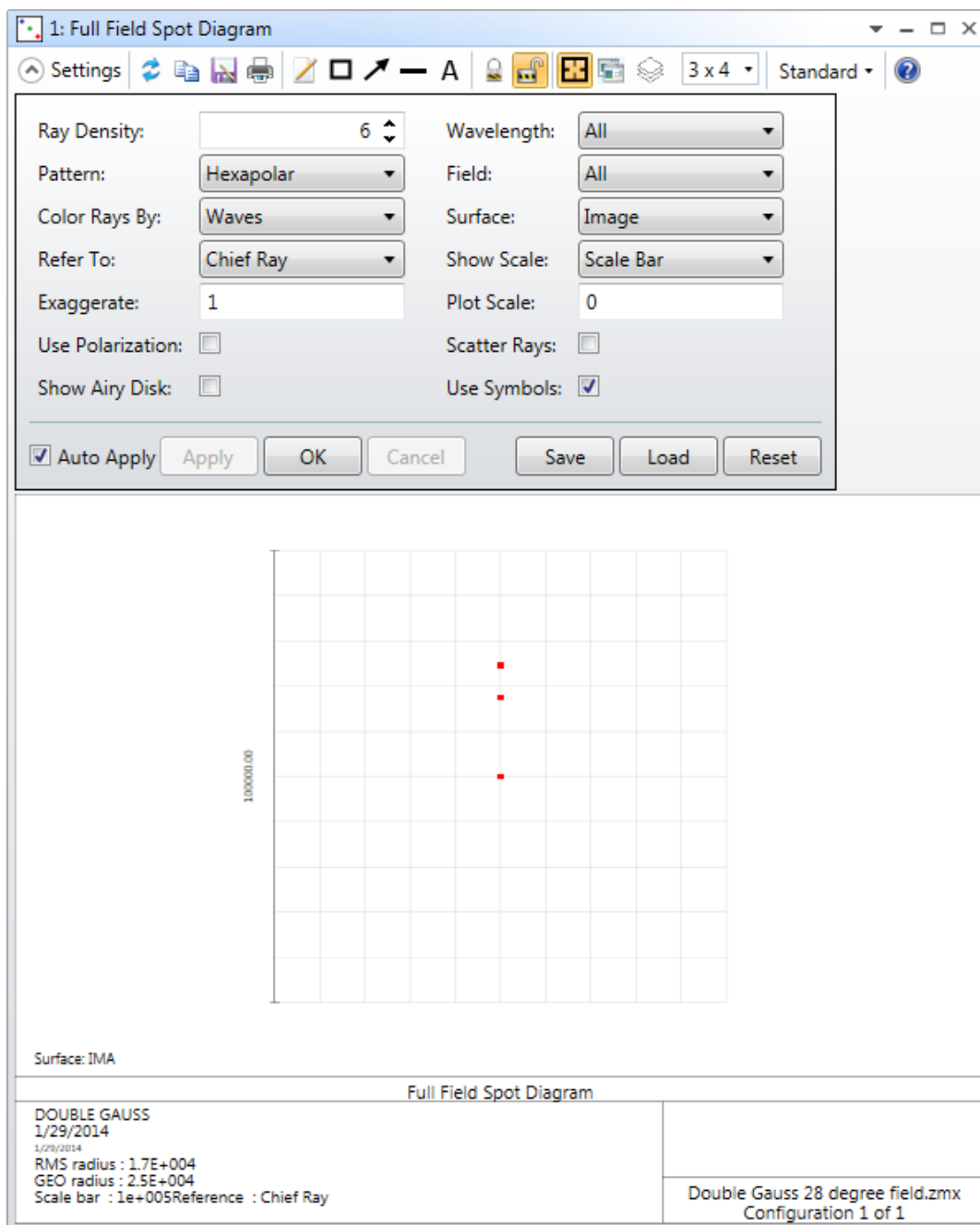
Most of the options are identical to the Standard Spot diagram.

Exaggerate This dimensionless factor magnifies the transverse aberrations to make the spot structure more apparent, although clearly there is an inherent distortion in using this feature. If "chief ray" is selected as the reference point, then the chief ray for field position 1 will be used.

Discussion The "Full Field" spot diagram type is similar to the "Standard" type, except all of the spots are plotted with respect to the same reference point, as opposed to a separate reference point for each field position. This provides some

---

idea of how the spot would look relative to the other field points. For example, this can be used to determine if two closely spaced image points can be resolved. The "Full Field" spot diagram type is less useful if the spot radius is small compared to the total field size, because in this case the spots for each field will appear very small compared to the distance between the spots. To increase the detail of the spots, enter a factor for the **Exaggerate** setting.

Matrix Spot Diagram

Show spot diagram as a matrix of individual diagrams, with each field along a row and each wavelength down a column.
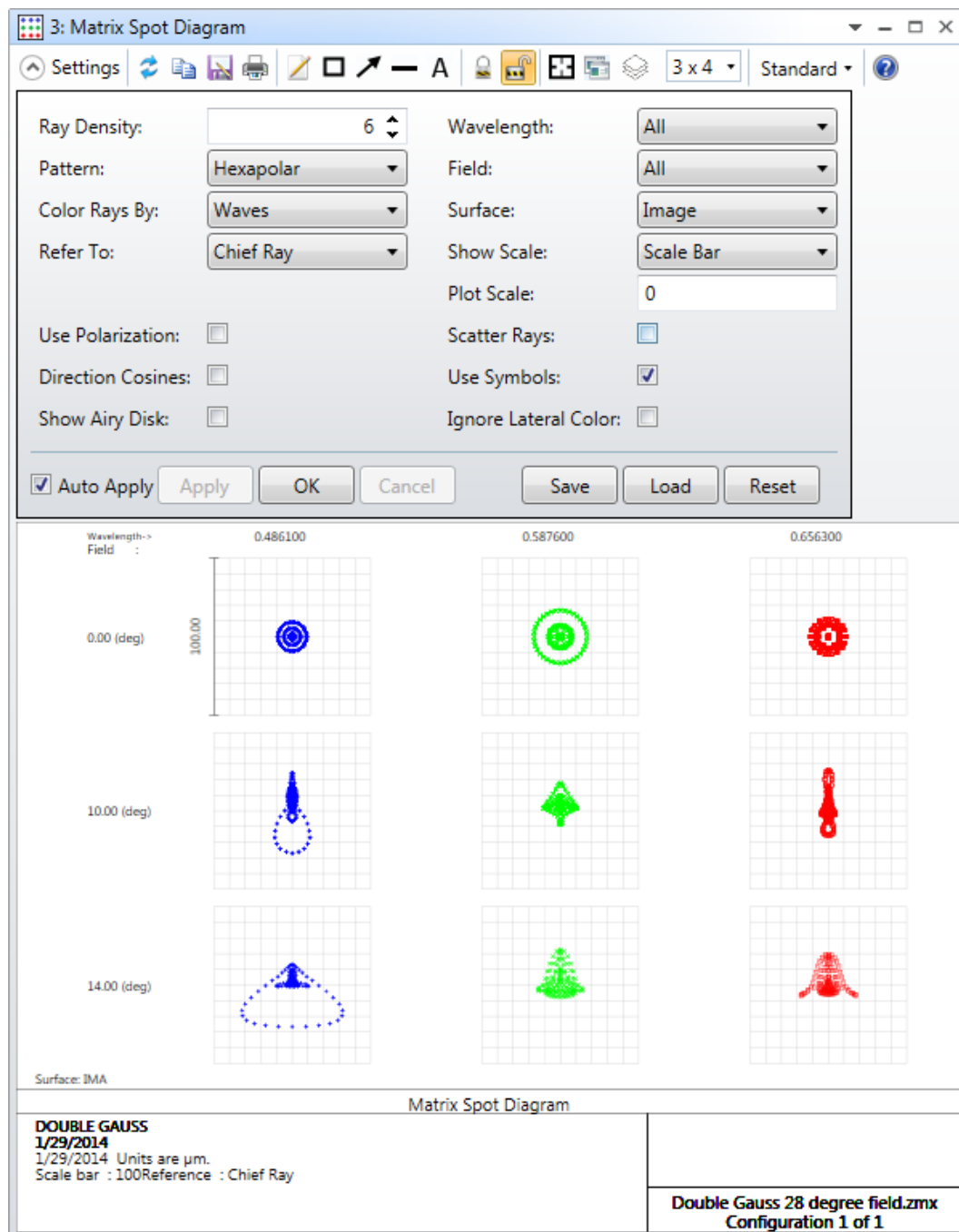
The options for the matrix spot diagram are similar to those for the standard spot diagram, except for the addition of the following option.

Ignore Lateral Color If checked, this option will reference each spot diagram to the reference point for each field and wavelength independently. This in effect ignores the effects of lateral color which can displace the reference points for each wavelength.

Discussion The matrix representation is a convenient way of distinguishing the components of wavelength dependent aberrations.

Configuration Matrix Spot Diagram

Show spot diagram as a matrix of individual diagrams, with each field along a row and each configuration down a column.
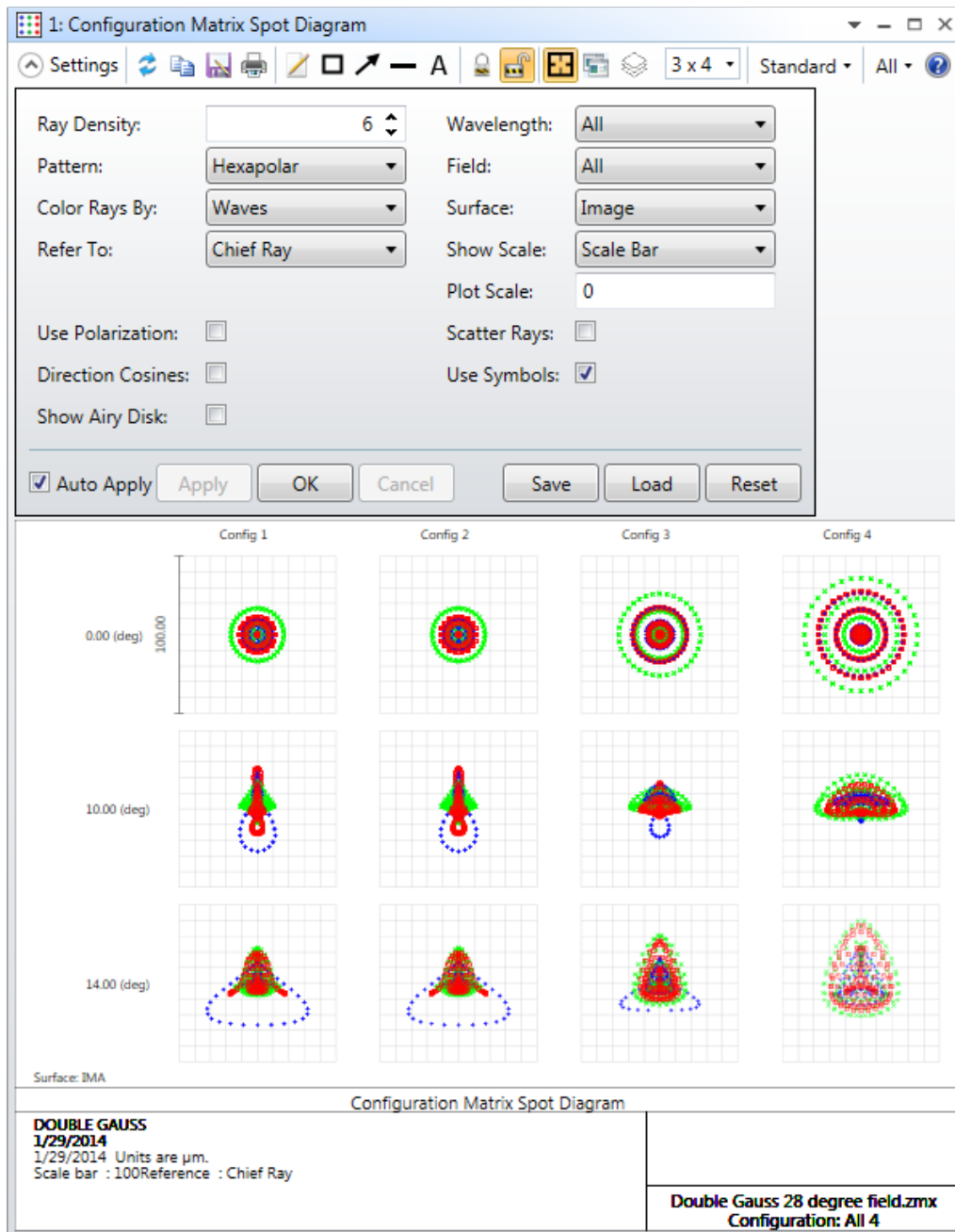
The options for the configuration matrix spot diagram are similar to those for the standard spot diagram.

Discussion The configuration matrix representation is a convenient way of distinguishing the components of configuration dependent aberrations. Along the left side of the plot the field positions are listed; only the field positions for the last configuration are listed if more than one configuration is shown and the field definitions are part of the multi-configuration data which changes.

Cardinal Points

This feature lists for the selected range of surfaces and wavelengths the locations of the principal, nodal, anti-nodal, and focal planes. The calculation is done for any defined wavelength and either the X-Z or Y-Z orientation.

First Surface   The starting surface number of the group to compute the cardinal points for.

Last Surface   The ending surface number of the group to compute the cardinal points for.

Wavelength   The wavelength number to use for the computation.

Orientation   The orientation to use for computing the cardinal plane locations.

Discussion   This feature may not return reliable results if coordinate breaks or non-centered optics are included within the specified surface range.

Y-Ybar Drawing

The Y-Ybar diagram is a plot of marginal ray height as a function of chief ray height for a paraxial skew ray at every surface in the lens.
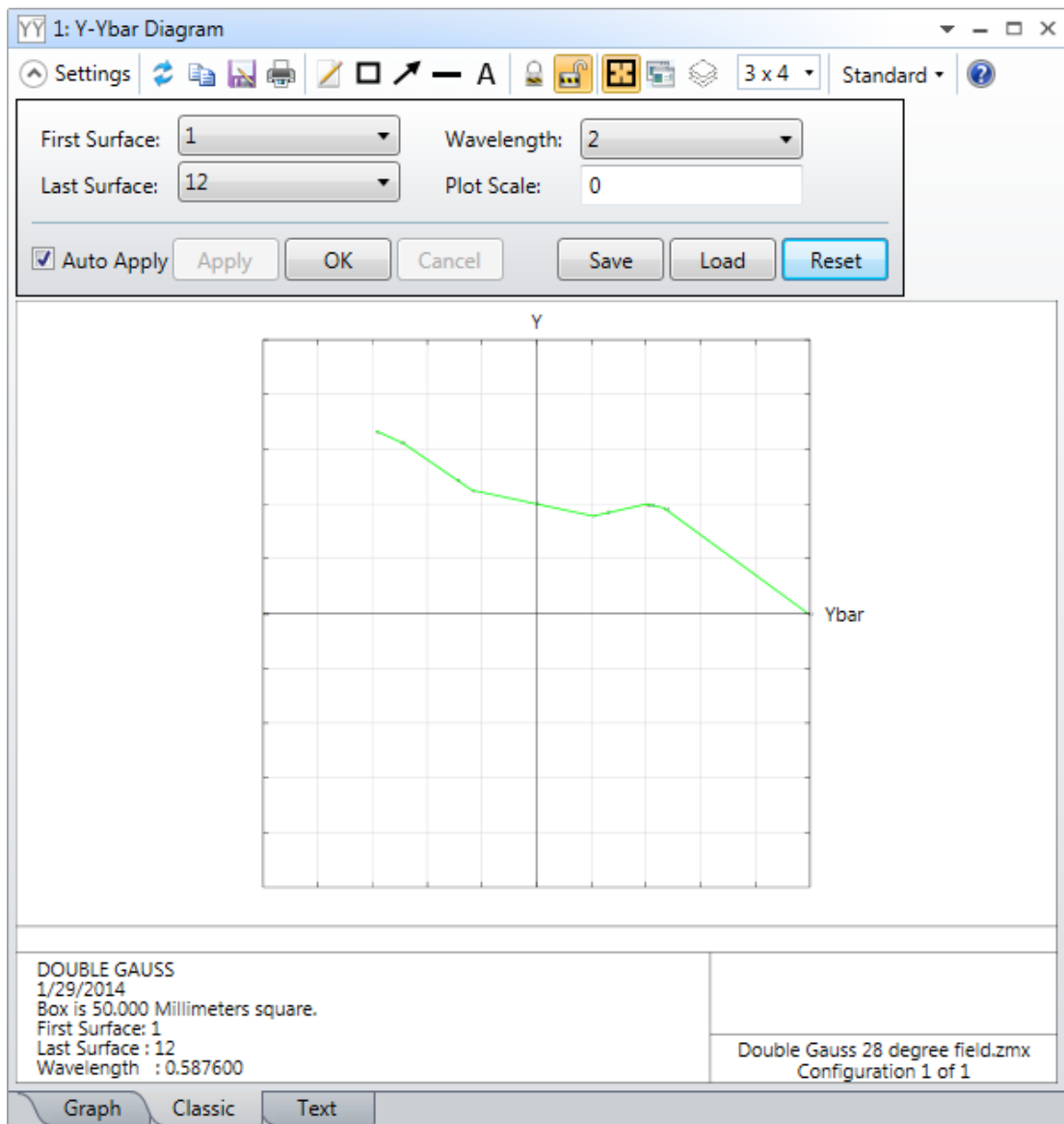
First Surface   The first surface for which data will be plotted.

Last Surface   The last surface for which data will be plotted.

Wavelength   The wavelength number to be used in the calculation.

Plot Scale   Sets the maximum scale for the plot. The plot is always shown in a square box; the default scale is the maximum transverse ray coordinate. Enter zero for automatic.

Vignetting Plot

Calculates fractional vignetting as a function of field angle
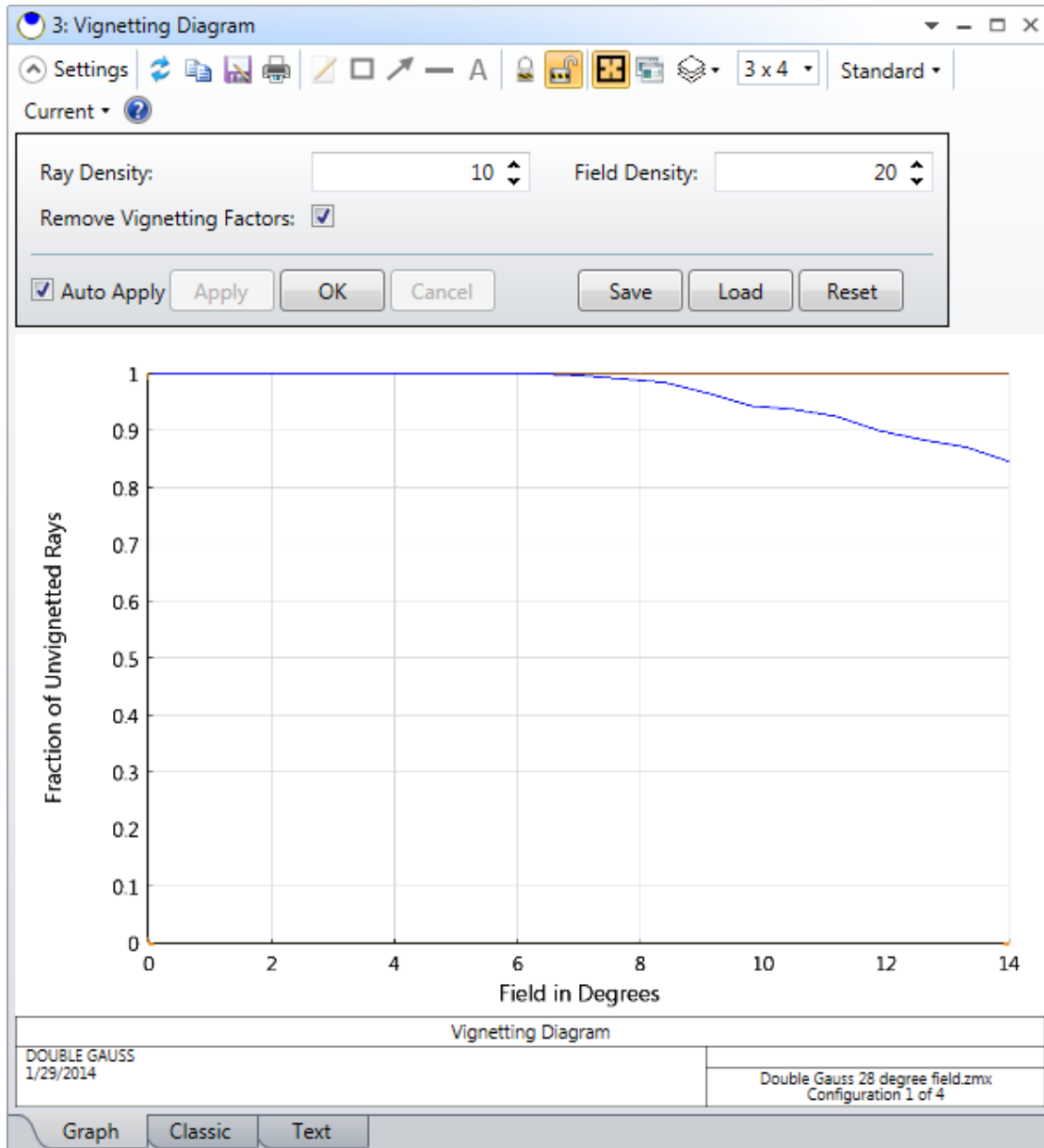
Ray Density The ray density specifies the number of rays to be traced. The more rays traced, the greater the accuracy, although the computation time increases. For a ray density of n, Zemax traces a grid of (2n+1) x (2n + 1) rays at each field point.

Field Density The field density is the number of points between zero degrees and the maximum field angle specified at which vignetting is calculated, intermediate values are interpolated.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Discussion Fractional vignetting is the percentage of rays incident upon the entrance pupil which pass all obscurations and apertures in the system and survive to the image surface, normalized to relative pupil area. The graphic generated by

this function shows fractional vignetting as a function of field position. If too few rays are used, the results may be inaccurate. This is especially true in systems with many apertures and large field angles.

Only the primary wavelength is used in this calculation. This is a geometric calculation. Only positive y field positions are used, therefore this feature is only appropriate for rotationally symmetric lenses and fields. Rays which cause errors such as missing a surface or those which are TIR are considered vignetted.

See also "Relative Illumination"

Comment about vignetting factors

Vignetting factors determine the size and shape of the pupil as seen from different field points (see "Vignetting factors" for a full discussion). Because this analysis feature needs to trace rays at arbitrary intermediate field points where no specific vignetting factors are defined, the use of vignetting factors is not recommended. If "Remove Vignetting Factors" is checked on (the default), any defined vignetting factors will automatically be replaced with surface apertures for this computation. The surface aperture method is generally more accurate than the vignetting factor method when the pupil is overfilled with light. The resulting data may be different between the two methods. In some cases, particularly where the vignetting factors are being used to define the shape of the source beam rather than the apertures of the optics, it may be required to use the defined vignetting factors. In this case, check the "Remove Vignetting Factors" box off. Zemax will then use the closest defined field to determine the vignetting factors to use for an arbitrary field point.

Incident Angle vs. Image Height

This feature computes the incident angle on the image surface of the lower marginal, chief, and upper marginal rays as a function of image height.
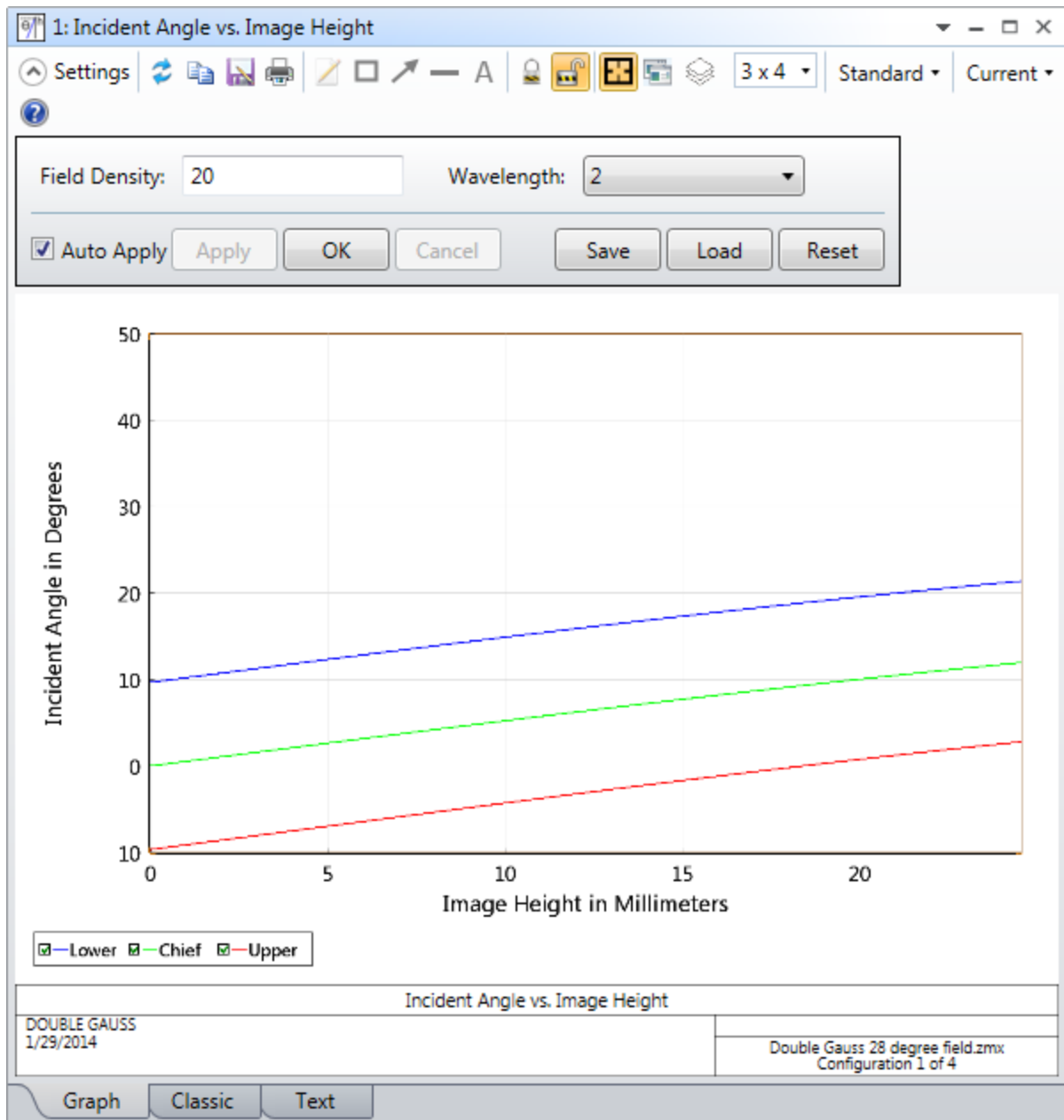
Field Density The number of points in the field, from 0 to the maximum +y field coordinate.

Wavelength The wavelength number to use for the analysis.

Discussion This feature traces three rays: one at the bottom of the entrance pupil, the chief ray, and another at the top of the entrance pupil. The (signed) angle of incidence on the image surface (which is assumed to be flat) is computed for each of these three rays, as the field coordinate changes from 0 to the maximum +y field. The image height based upon the chief ray y-intercept coordinate is also computed. The feature then plots (or tabulates as text) the three resulting angles as a function of the resulting image height.

This feature is only intended to be used for rotationally symmetric systems with no vignetting, and may not produce meaningful results for other types of systems.

Aberrations
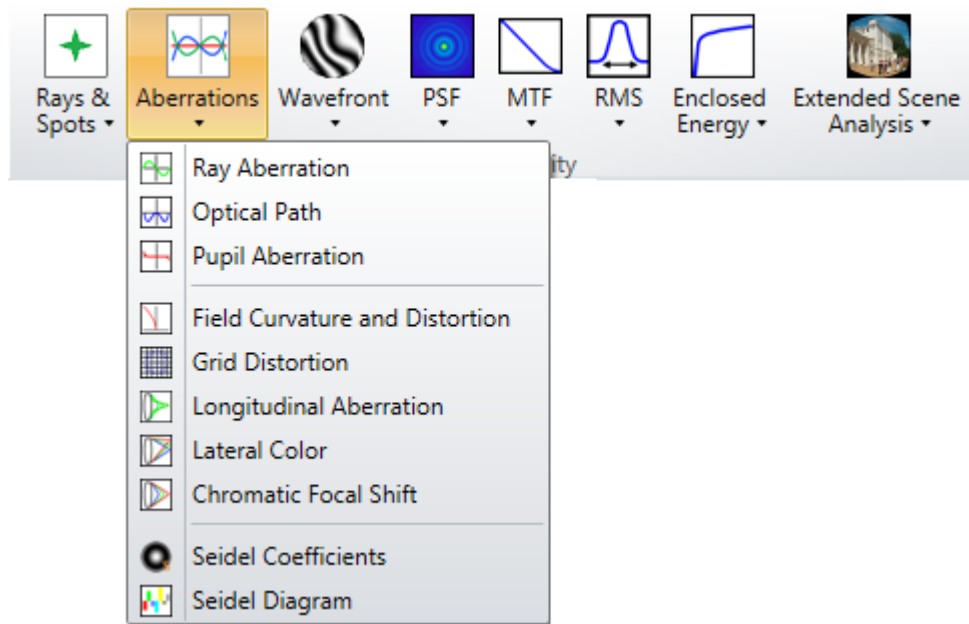
Various views of the departure of the performance of the optical system from the predictions of paraxial optics.

Ray Aberration

Shows ray aberrations as a function of pupil coordinate.

Plot Scale Sets the maximum vertical scale for the plots. The maximum scale is in micrometers for ray fans, waves for OPD plots, or percent for entrance pupil aberration plots. This overrides the automatic selection of scale for the plots. Enter zero for automatic scaling.

Number of Rays This is the number of rays traced on each side of the origin of the plot.

Wavelength The wavelength number for which the calculation should be performed.

Field The field number for which the calculation should be performed.

Tangential Selects which aberration component to plot for the tangential fan. Since tangential fans are functions of the y pupil coordinate, the default is to plot the y component of the aberration.

Sagittal Selects which aberration component to plot for the sagittal fan. Since sagittal fans are functions of the x pupil coordinate, the default is to plot the x component of the aberration.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves.

Check Apertures Specifies whether or not to check if rays pass all surface apertures. If this is selected, rays which do not pass surface apertures will not be drawn.

Vignetted Pupil If checked, the pupil axis will be scaled to the unvignetted pupil, in which case the data will reflect the vignetting in the system. If unchecked, the pupil axis will be scaled to fit the vignetted pupil.

Surface Selects the surface at which the fan is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Discussion

The tangential fans show either the x or the y component of the transverse ray aberration as a function of the y pupil coordinate of the ray. The default option is to plot the y component of the aberration. However, since transverse ray aberrations are vectors, this is an incomplete description of the aberration. When Zemax plots the y component, the plot is labeled EY, when plotting the x component of the aberration, the plot is labeled EX.

The vertical axis scale is given at the bottom of the graph. The data being plotted is the difference between the ray intercept coordinate and the chief ray intercept coordinate. The tangential fan is the plot of the difference between the ray x or y coordinate and the chief ray x or y coordinate at the primary wavelength, as a function of the y pupil coordinate. The sagittal plot is the difference between the ray x or y coordinate and the chief ray x or y coordinate as a function of the x pupil coordinate. The horizontal scale for each graph is the normalized entrance pupil coordinate, either PX or PY.

If "All" wavelengths are shown, then the plot is referenced to the primary wavelength chief ray. If monochro- matic, then the chief ray for the selected wavelength is used as a reference. For this reason, the data for non- primary wavelengths will in general change when switching between monochromatic and polychromatic displays.

Because ray aberrations are vectors, with both an x and a y component, the ray aberration fan is an incomplete description of the aberrations, especially when the image surface is rotated or the system is otherwise non- rotationally symmetric. Also, the fans only indicate aberrations along two "slices" through the pupil, rather than over the entire entrance pupil. The primary purpose of the ray fan plot is to determine what aberrations are present in the system; it is not a complete description of the system performance, especially for systems without rotational symmetry.

Evaluating results at intermediate surfaces

Zemax can compute analysis results for surfaces other than the image surface by applying assumptions. These assumptions work in most cases, however, there may be cases where the methods described here for analyzing intermediate surfaces are not appropriate, including systems which require ray aiming. All of the changes made to the lens described are made on a copy of the original lens data made for analysis purposes so no changes are made to the original lens data.

If the field type is either real or paraxial image height, the field type is changed to angle or object height for infinite or finite conjugate systems, respectively. The angles and heights used correspond to the primary wavelength chief ray angles and heights as computed for the unaltered system.

If the selected surface precedes the stop surface, Zemax moves the stop surface to a (possibly virtual) dummy space prior to the existing surface 1. Unless the system aperture is object space numerical aperture or cone angle, the system aperture is changed to entrance pupil diameter, and the aperture value is set equal to the original paraxial entrance pupil diameter computed for the original stop position. Note this assumption might not be valid for systems that require ray aiming. If the selected surface follows the stop surface, no changes are made to the system aperture or stop definitions.

Surfaces which follow the selected surface are then deleted. The glass of the new image surface is set to be the same as the selected surface.

Most analysis features that compute results for focal mode systems make more sense if the rays are allowed to come to a focus after refraction from the desired surface. For example, the OPD plot is generally a useful diagnostic only when the OPD is measured at the (possibly virtual) focus of that surface. Other features which require a temporary image to be formed include PSF, MTF, and diffraction encircled energy. For these features requiring a temporary image, the new image surface is set to be a standard plane surface. A paraxial marginal ray height solve is placed on the selected surface thickness to place the image surface at paraxial focus for the selected surface. The analysis computation then proceeds at this newly created intermediate image surface. Note the analysis occurs at the paraxial focus formed by the rays after refracting through the selected surface. This shift to paraxial focus is not performed if the intermediate system is in afocal mode. For more information on afocal mode see "Afocal Image Space".

Some analysis features make more sense if the data is evaluated directly on the surface itself, without allowing the rays to focus to an image. These features include the various spot diagrams, footprint analysis, geometric encircled energy, line/edge spread function, and extended source encircled energy.

Optical Path

Shows optical path difference as a function of pupil coordinate.

The options are identical to those for ray aberration fans, except the only option for "Tangential" and "Sagittal" is OPD, since OPD is a scalar quantity.

Discussion The vertical axis scale is given at the bottom of the graph. The data being plotted is the optical path difference, or OPD, which is the difference between the optical path length of the ray and the optical path length of the chief ray. Usually, the calculation is referenced back to the difference between the ray path lengths at the system exit pupil. The horizontal scale for each graph is the normalized entrance pupil coordinate.

If "All" wavelengths are shown, then the plot is referenced to the primary wavelength based reference sphere and chief ray. If monochromatic, then the reference sphere and chief ray for the selected wavelength is used as a reference. For this reason, the data for non-primary wavelengths will in general change when switching between monochromatic and polychromatic displays.

Pupil Aberration

Shows entrance pupil distortion as a function of pupil coordinate.
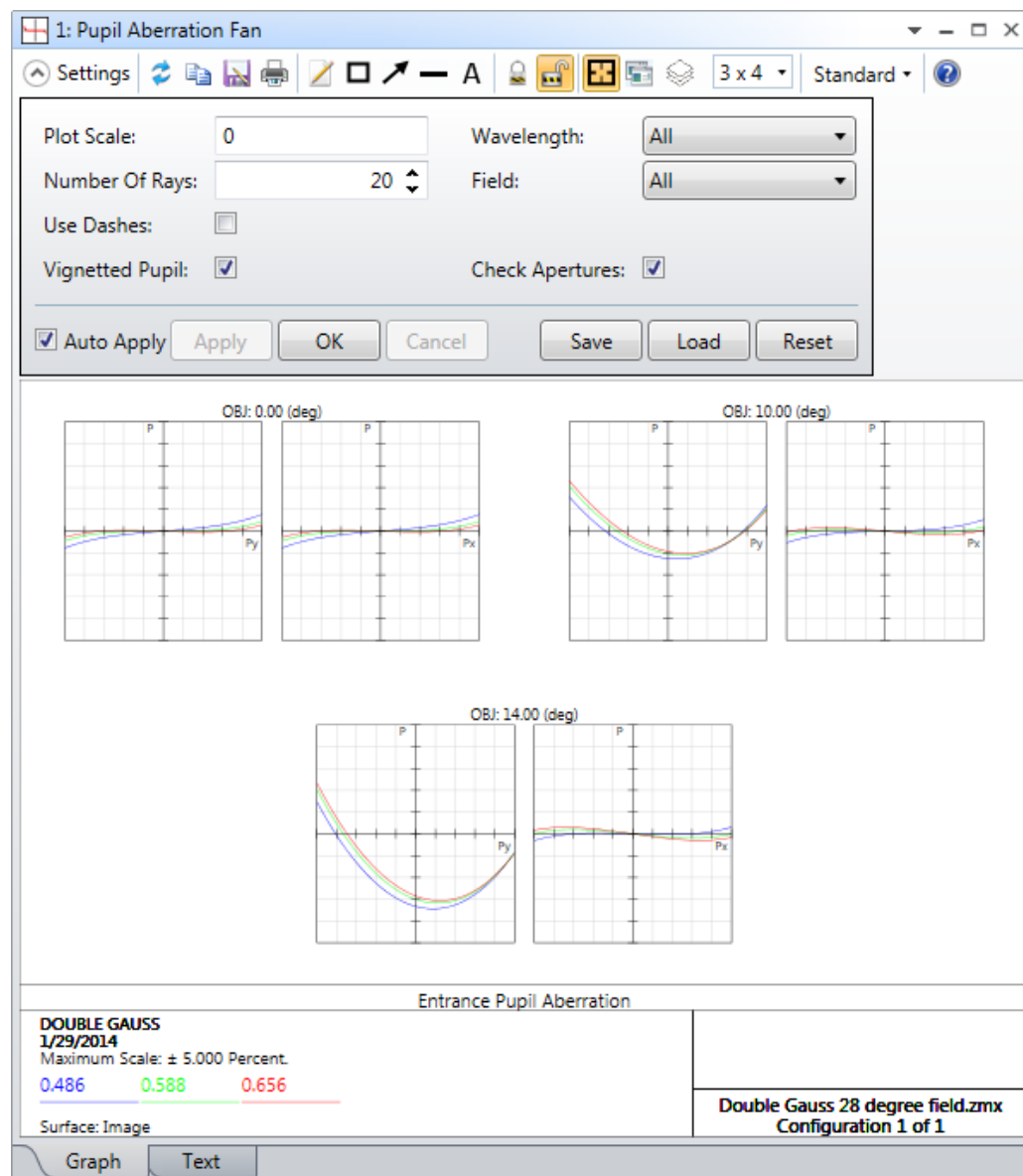
The options are identical to those for ray aberration fans, except the only option for "Tangential" and "Sagittal" is pupil aberration, since pupil aberration is a scalar quantity. The "Surface" only allows "Image" because this data is always computed at the stop surface.

Discussion Entrance pupil aberration is defined as the difference between the real ray intercept on the stop surface and the on axis primary wavelength paraxial ray intercept as a percentage of the paraxial stop radius. If the maximum aberration exceeds a few percent, ray aiming (see "Ray Aiming") may be required to get the rays in object space to correctly fill the stop surface. If ray aiming is switched on, the entrance pupil aberration will appear to be zero (or a very small residual value), because the distortion is accounted for by the ray tracing algorithm. This is used as a check that ray aiming is working correctly. The definition used here for pupil aberration is not intended to be complete or in agreement with other definitions. The sole purpose of this feature is to provide guidance as to whether or not ray aiming is required.

Field Curvature and Distortion

Displays the field curvature and distortion plots.
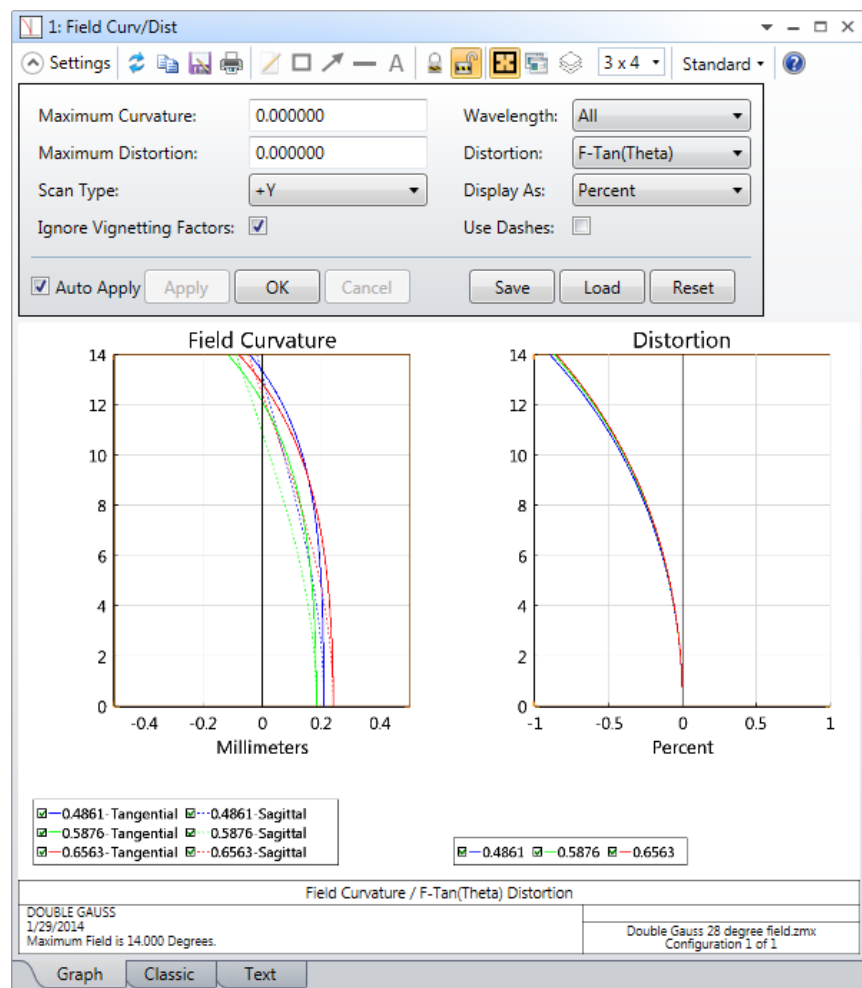
Max Curvature Defines the maximum scale for the field curvature plot in lens units. Enter zero for automatic.

Max Distortion Defines the maximum scale for the distortion plot. Enter zero for automatic.

Wavelength The wavelength number to be used in the calculation.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Ignore Vignetting Factors See the discussion section.

Distortion Select F-Tan(theta), F-Theta, Calibrated F-Theta, or Calibrated F-Tan(theta) distortion. See the discussion section for details.

Display As Selects whether to display distortion as a percentage or as an absolute length.

Scan Type Choose +y, +x, -y, or -x field scan direction.

Discussion:

The field curvature plot shows the distance from the image surface to the paraxial image surface as a function of field coordinate. The tangential data are the distances measured along the Z-axis from the image surface to the paraxial image surface measured in the tangential (YZ) plane. The sagittal data are the distances measured in the plane orthogonal to the

---

tangential plane. The base of the plot is on axis, and the top of the plot represents the maximum field (angle or height). There are no units on the vertical scale because the plot is always normalized to the maximum field coordinate along the scan direction.

The field scan is along the +Y field by default. If +X or -X is selected, then the field scan is along the X direction, in which case the tangential curve is for the XZ plane, and the sagittal curve is in the YZ plane.

The field curvature plot does not always start at zero for zero field. The reason is that the plot shows the distance from the image surface to the paraxial image surface, and the image surface need not be coincident with the paraxial image surface. If there is any defocus, then the two surfaces are offset, and so is the field curvature data.

Distortion in percent is defined as the real chief ray height, minus the reference ray height, divided by the reference ray height, times 100:

$$Distortion = 100 \times \frac{y_{chief} - y_{ref}}{y_{ref}}$$

where all heights are taken to be the image surface radial coordinate, at whatever image surface is defined (the data is not referred to the paraxial image surface). The reference ray height is computed by tracing a real ray from a very small field height, and then scaling the results as required. This generalization permits the computation of reasonable distortion values even for systems not well described by paraxial ray tracing. For F- Tan(theta) distortion, the reference height for the undistorted ray in a rotationally symmetric system at paraxial focus is given by

$$y_{ref} = f tan\theta$$

where f is the focal length and θ is the angle in object space. Note that this reference is meaningless for optical systems that use field angles at or greater than 90 degrees. If a system with field angles over 89 degrees is analyzed with this feature, the maximum computed field angle will be limited to 89 degrees.

F-Theta distortion does not use the tangent relationship, but instead uses the height given by the focal length of the lens multiplied by the angle the chief ray makes in object space. This so-called F-Theta height is only meaningful in systems with the object at infinity, when field heights are measured in angles. The reference height for the undistorted ray at the paraxial focal plane is given by

$$y_{ref} = f\theta$$

where f is the focal length and θ is the angle in object space. F-Theta is used in scanning systems where the image height needs to be linear with scan angle.

Calibrated distortion, whether F-Tan(theta) or F-Theta, is similar to the other distortion definitions, except that the "best-fit" focal length is used rather than the system focal length. Calibrated distortion measures the deviation between the reference and actual image height and selects the best f value to minimize the distortion, without the restriction that the proportion of linearity be defined by the focal length of the system. A focal length is chosen that best fits the data, rather than the system focal length, although in general the best fit focal length is close to the system focal length. The calibrated focal length used is given on the text listing for this feature.

One effect of this definition of calibrated distortion is the non-zero distortion at zero field angle. The reason for this non-zero distortion is best explained by looking at the limiting behavior of the definition of calibrated distortion. Calibrated distortion in percent is defined as

$$Distortion = 100 \times \frac{y_{chief} - y_{ref}}{y_{ref}}$$

For small angles, the real y chief ray coordinate in any reasonable optical system is well described by $y_{chief} = f\theta$ and the reference ray coordinate by $y_{ref} = f'\theta$ where f' is the calibrated focal length, so the distortion near zero field is

$$Distortion = 100 \times \frac{y_{chief} - y_{ref}}{y_{ref}} = 100 \times \frac{f\theta - f'\theta}{f'\theta} = 100 \times \frac{f - f'}{f'}$$

which is not zero, unless $f = f$. Therefore, the calibrated distortion is not generally zero on axis. This does

not mean the image height is not zero, it is an artifact of the different choice of focal lengths for the reference and actual ray coordinates near the optical axis. Note that a percentage distortion near the axis is not significant because the field itself approaches zero at zero field angle.

For non-rotationally symmetric systems, distortion is poorly defined and the data presented is probably meaningless. The reason is that no single number adequately describes distortion at a single field point if the system is not rotationally symmetric. Instead, see "Grid Distortion".

Strictly speaking, the field curvature and distortion plots are only valid for rotationally symmetric systems with plane object and image surfaces. However, Zemax uses a generalization of the field curvature and distortion concepts to give reasonable results for some, but not all, non-rotationally symmetric systems. Caution should be used when interpreting these data for non-rotationally symmetric systems or systems with object and/or image surfaces that are not planes.

By default, Zemax ignores vignetting factors when computing the field curvature and distortion plots. Vignetting factors can change the chief ray location on the stop surface, such that the chief ray no longer goes through the center of the stop.

Computing distortion when using real image height as the field type

The distortion cannot be computed in the manner described here if the field type is real image height. The reason is that when using real image height, Zemax iterates each chief ray trace to find the exact object space angle to hit the desired image coordinate. Because the desired image coordinate is always reached, the image height is linear with field coordinate. The iteration is thus implicitly removing the distortion. Instead, Zemax automatically changes the field type from real image height to field angle for the purposes of this computation. The field angles at each defined field point are then chosen to yield the identical real image height coordinates. Using this method, the deviation from linearity is now retained and the distortion may be computed accurately. If the optical system is not axial symmetric, the conversion from real image height to field angle may yield unexpected results as the field angle scan direction and the real image coordinate direction may no longer be coplanar, and care should be taken in interpreting the results. It is recommended that the system be converted to use field angle or object height rather than real image height for non-axial systems when evaluating distortion.

Grid Distortion

Displays a grid of chief ray intercept points to indicate distortion.
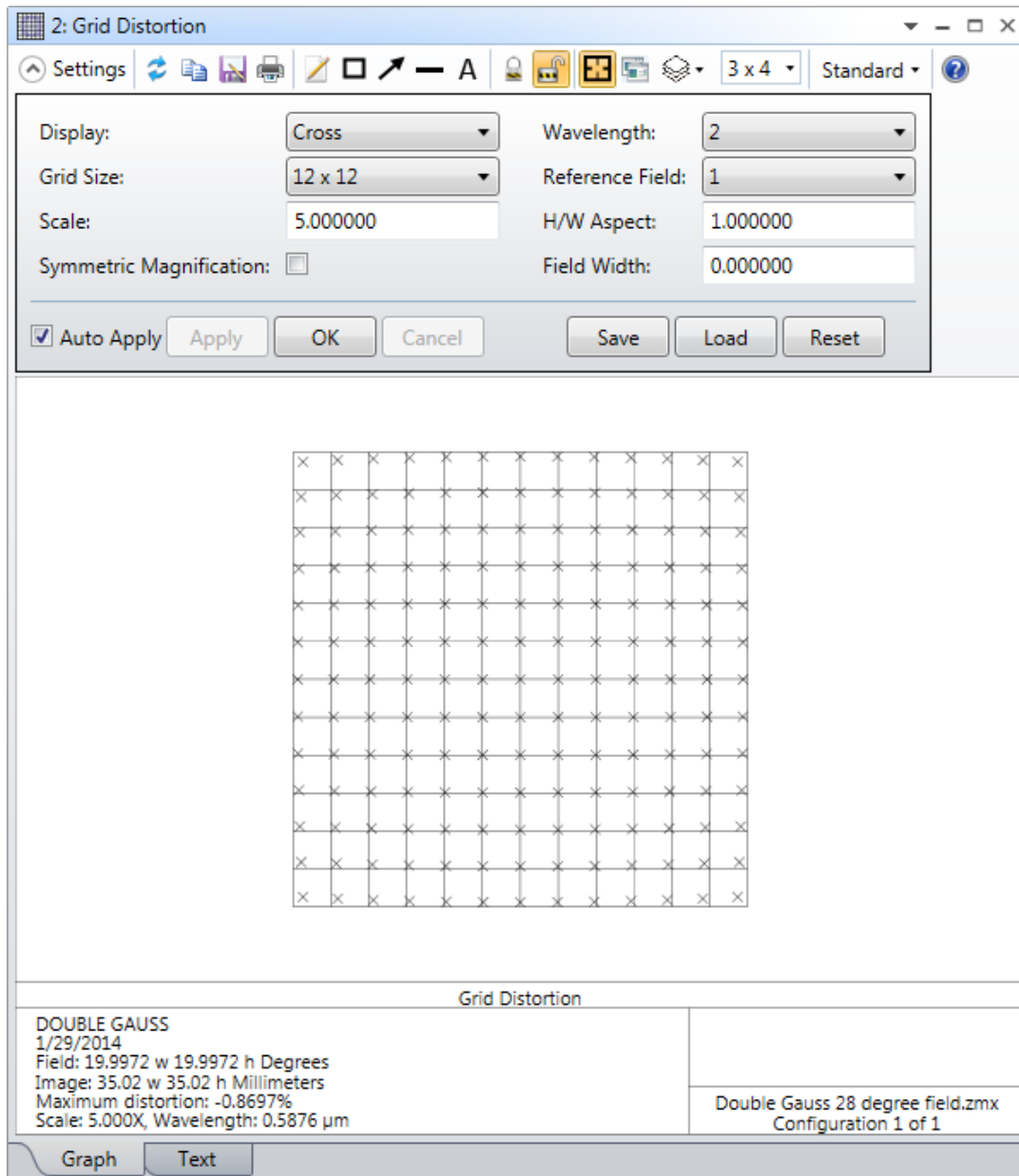
Display Select either "Cross" to mark each chief ray intercept with a cross, or "Vector" to plot a vector from the ideal image point to the actual chief ray image point.

Grid Size The size of the grid.

Wavelength The wavelength number to be used in the calculation.

Ref. Field The reference field position. See the "Discussion".

Scale If the scale is different than 1.0, then the "x" points on the distortion grid will be exaggerated by the selected scale factor. The scale factor may be negative to change positive to negative distortion on the plot.

H/W Aspect If unity, then a square field will be selected. The output image may not be square if the system is not symmetric, but the object field will be square. If the H/W aspect is greater than 1, then the "height" or "y" field will be expanded by the aspect ratio. If the aspect is less than 1, then the "height" or "y" field will be compressed by the aspect

ratio. The aspect ratio is the "y" field height divided by the "x" field width. The aspect ratio only affects the input field; the image aspect ratio is determined by the optical system imaging properties.

Symmetric Magnification If checked, then the X magnification is required to be identical to the Y magnification. This causes distortion to be referenced to a symmetric predicted grid rather than an anamorphic predicted grid.

Field Width The full width, in field units, of the "x" field of view. If the current field type is real image height, Zemax will temporarily change the field height to field angle, and the "Field Width" value will be in angular units. See "Computing distortion when using real image height as the field type".

Discussion:

This feature displays or computes the coordinates of a grid of chief rays. In a system without distortion, the chief ray coordinates on the image surface follow a linear relationship with the field coordinate:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} f_x \\ f_y \end{bmatrix}$$

where xp and yp are the predicted image coordinates relative to a reference image point and fx and fy are the linear coordinates on the object surface relative to a reference point. For optical systems using angles as a field definition, fx and fy are the tangents of the field angles (the field coordinates must be linear, therefore, tangents of angles rather than angles are used). To compute the ABCD matrix, Zemax traces rays over a very small region centered upon the reference field position. Usually, this is the center of the field of view. Zemax allows selection of which field position to use for reference.

By default, Zemax sets the corner of the field grid in object space to be at the maximum radial field distance. Because object height is linear with the tangent of the field angle, the full width of the field when angles are used to define the field is given by

$$\theta_{wide} = 2\, tan^{-1} \left[\frac{\sqrt{2}}{2} \tan \theta_r \right]$$

where θ_r is the maximum radial field angle at the corner of the field.

The ray coordinates in image space for the very small field of view are used to determine the ABCD matrix components. The use of an ABCD matrix allows for coordinate rotations. If the image surface is rotated, such that a y object coordinate images to both an x and a y image coordinate, the ABCD matrix will automatically account for the rotation. The grid distortion plot shows the linear grid, and then marks the actual chief ray intercept for a ray with the same linear field coordinates with an "X" for each point on the grid.

If the optical system is not rotationally symmetric, then the distortion is not generally radial. The distortion is a vector, and the magnitude of the distorted vector must be used to compute the total distortion.

The text listing available tabulates the predicted image coordinate, the actual image coordinate, and the "percent distortion" defined by

$$P = 100\% \, \frac{R_{distorted}}{R_{predicted}} \text{ where}$$

$$R_{real} = \sqrt{(x_r)^2 + (y_r)^2},$$

$$R_{predicted} = \sqrt{(x_p)^2 + (y_p)^2} \text{ and}$$

$$R_{distorted} = \sqrt{(x_p - x_r)^2 + (y_p - y_r)^2}$$

and the subscripts r and p refer to the real and predicted coordinates on the image surface relative to the reference field position image location, respectively. Because the values for R_{real} and $R_{predicted}$ are always positive, this definition will always yield a positive value for P. However, it is still a frequently useful concept to distinguish between "positive" and "negative" distortion. To support this, if R_{real} is less than $R_{predicted}$, then the sign of P is changed to be negative.

NOTE: This definition of generalized distortion may not be applicable in all cases, and the results should be used with caution.

SMIA-TV Distortion

The text listing of this feature also computes an alternate mea- sure of distortion called Standard Mobile Imaging Architecture TV (SMIA-TV) Distortion. SMIA-TV Distortion is defined as

$$P_{TV} = 100\% \; \frac{A - B}{B}$$

where A = 0.5*(A1+A2) and A1, A2, and B are distances measured between three pairs of points in a rectangular field of view as shown in the image.
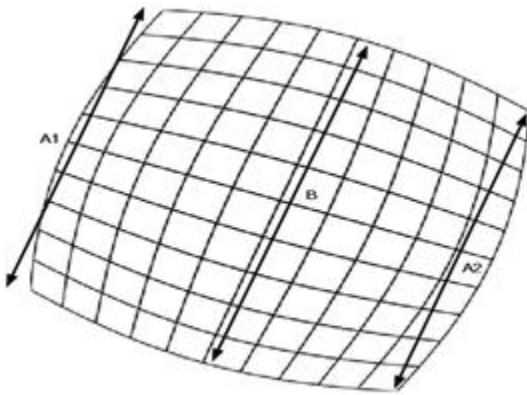
 The A distances are mea- sured between the top and bottom corners of the field of view at the left and right sides of the image, and the B distance is mea- sured between the top and bottom of the image at the x-field center of the image. If any one of the 6 required rays cannot be traced, the SMIA-TV distortion will be listed as zero.

Limitations of Grid Distortion

Zemax cannot compute the distortion in the manner described here if the field type is real image height. Instead, Zemax automatically changes the field type from real image height to field angle for the purposes of this computation. For more information see "Computing distortion when using real image height as the field type".

Grid distortion cannot be calculated if the field units are angles and the maximum angle equals or exceeds 90 degrees. This limitation is due to the assumption that the predicted image height is proportional to the tangent of the field angle in object space. When the field angle exceeds 90 degrees, the tangent does not predict the linear image height correctly.

The grid distortion plot is oriented in field units, so that the +y direction corresponds to the +y field coordinate in field units (see "Fields"). This may or may not be different than the orientation of a spot diagram or other plot plotted in image space units.

Longitudinal Distortion

Displays the longitudinal aberration as a function of pupil height at each wavelength.
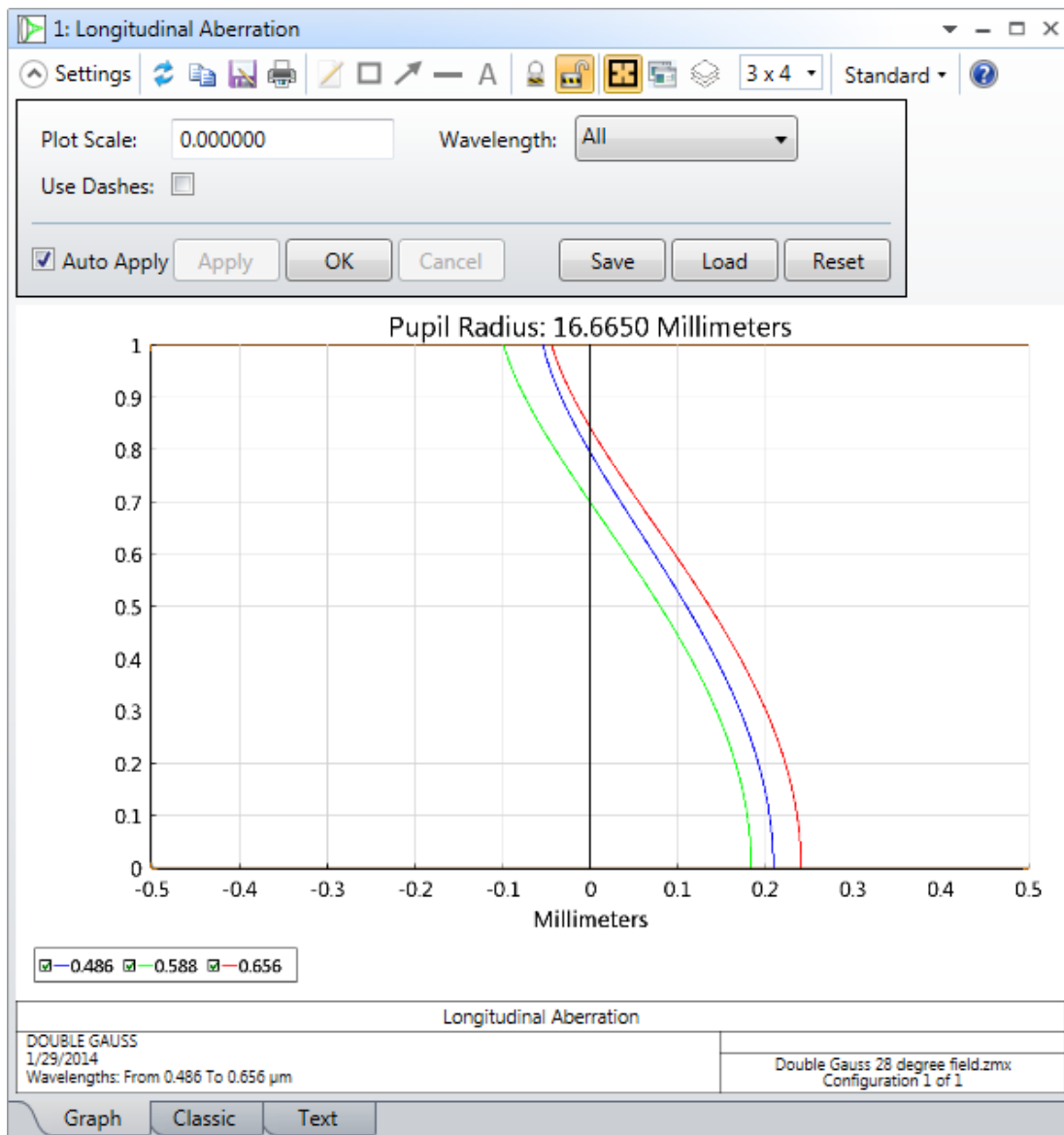
Plot Scale Defines the maximum scale for the plot in lens units. Enter zero for automatic.

Wavelength The wavelength number to be used in the calculation.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves.

Discussion This feature computes the distance from the image surface to where a zonal marginal ray "focuses", or crosses the optical axis. The computation is performed only for the on axis field point, and only for zonal marginal tangential rays as a function of pupil zone. The base of the plot is on axis, and the top of the plot represents the maximum entrance pupil radius. There are no units on the vertical scale because the plot is always normalized to the maximum entrance pupil radius. The horizontal scale is in lens units, and represents the distance from the image surface to the point where the ray crosses the optical axis. For afocal systems, the horizontal axis is diopters of power required to bring the rays to collimation.

Because longitudinal aberration is defined in terms of the distance to the ray-axis crossing point, this feature may produce meaningless data for non-rotationally symmetric systems. Great care should be exercised in interpreting this plot for non-rotationally symmetric systems.

Lateral Color

Displays the lateral color as a function of field height.
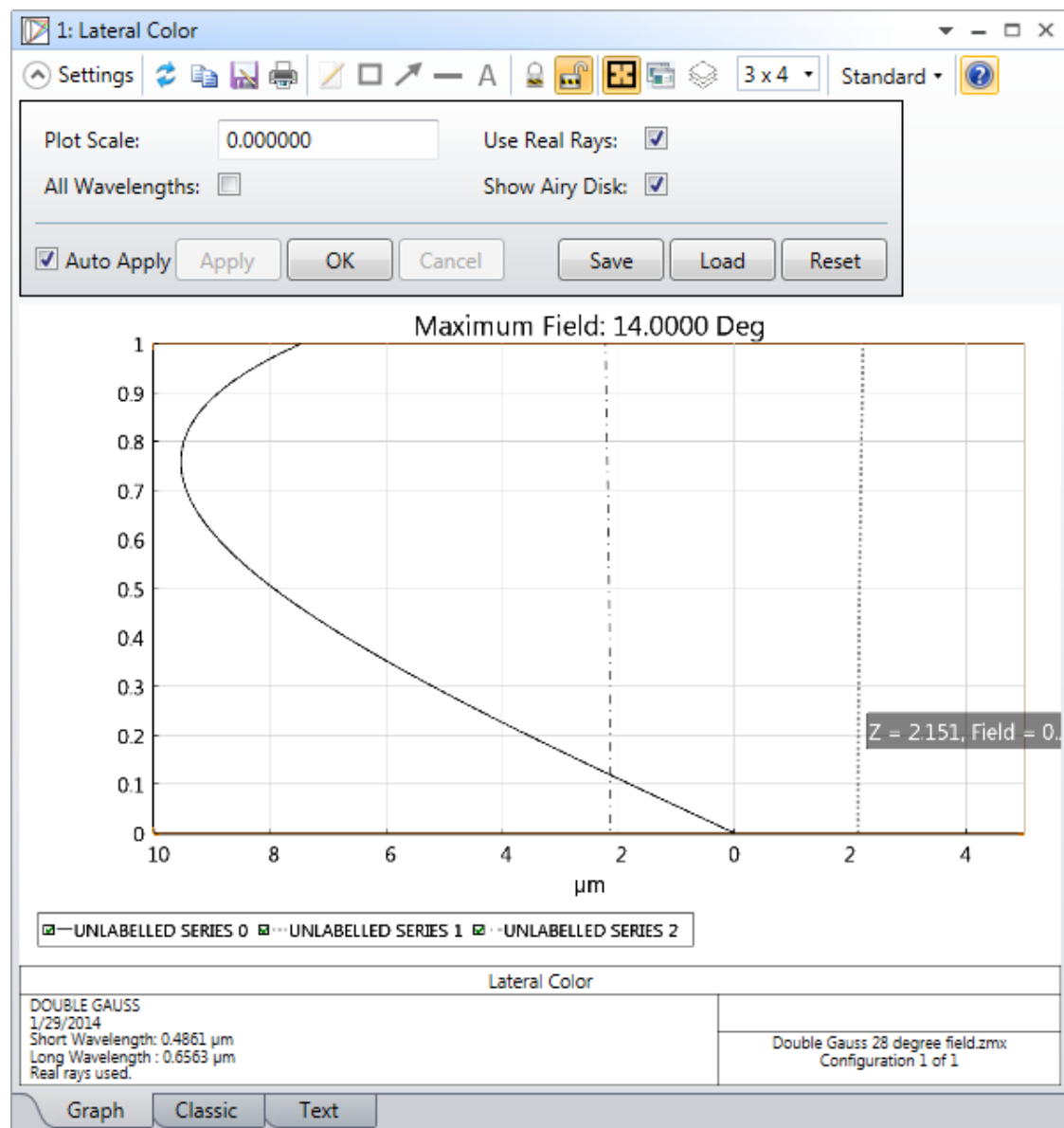
Plot Scale Defines the maximum scale for the plot in lens units. Enter zero for automatic.

Use Real Rays If checked, real rays are used, otherwise paraxial rays are used.

All Wavelengths If checked, then data for all defined wavelengths will be displayed. Each wavelength will be referenced to the primary wavelength. If not checked, then the difference between the shortest and longest wavelength rays will be used. See the discussion.

Show Airy Disk If checked, then the Airy disk radius at the primary wavelength will be plotted on either side of the reference line to indicate the extent of the Airy disk.

Chromatic Focal Shift

Shift in back focal length with respect to the primary wavelength
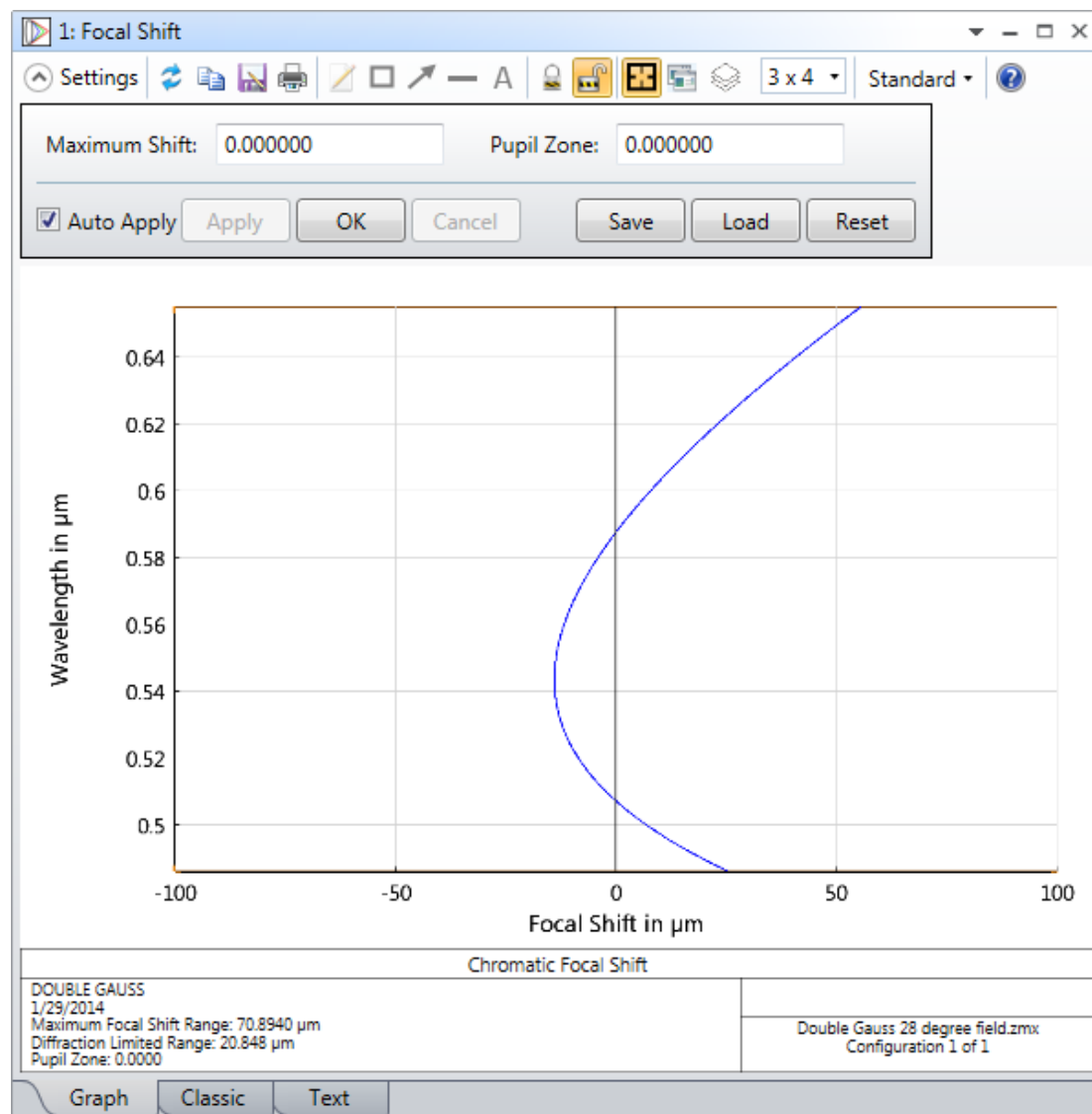
Maximum Shift The maximum extent in lens units for the horizontal axis. The vertical axis scale is set by the range of wavelengths defined. Enter zero for automatic.

Pupil Zone The radial zone in the pupil used to compute the back focus. The default value of zero means a paraxial ray will be used. Values between 0 and 1 mean real marginal rays are used in the appropriate zone in the entrance pupil. A 1 is at the edge of the pupil, or full aperture.

Discussion This is a plot of the shift in back focal length with respect to the primary wavelength. At each plotted wavelength, the shift in image space required to reach focus for that color marginal ray is computed. The shift distance is computed in the same media (glass or air) as the surface prior to the image surface. This plot may not be meaningful for non-paraxial systems.

The maximum shift setting overrides the default scaling. Units are in lens units. The entire plot is always referenced to the primary wavelength paraxial focus. The diffraction limited depth of focus listed is given by the formula $4\lambda F^2$, where F is the working F/# and λ is the primary wavelength

Seidel Coefficients

Displays Seidel (unconverted, transverse, and longitudinal), and wavefront aberration coefficients.

```
1: Seidel Coefficients                                                    ▼ _ □ ×

 ⌃ Settings  ⭍ 🗋 🖫 🖨  ╱ □ ↗ ─ A  ⌙ 🔒 🖽 🗗 ⬙  3 x 4 ▾  Standard ▾  ❓

  Wavelength: 2                              ▼

  ☑ Auto Apply  Apply    OK    Cancel      Save    Load    Reset
```

```
Listing of Aberration Coefficient Data

File : C:\Users\aaron.weleck\Documents\Zemax\Samples\Sequential\Objectives\Double Gauss 28 degree field.zmx
Title: DOUBLE GAUSS
Date : 1/29/2014


Wavelength                    :          0.5876 µm
Chief Ray Slope, Object Space :          0.2493
Chief Ray Slope, Image Space  :          0.2292
Marginal Ray Slope, Object Space:        0.0000
Marginal Ray Slope, Image Space :       -0.1675
Petzval radius                :       -623.2905
Optical Invariant             :          4.1551

Seidel Aberration Coefficients:

Surf    SPHA  S1     COMA  S2     ASTI  S3     FCUR  S4     DIST  S5     CLA (CL)     CTR (CT)
  1     0.114175    -0.008176     0.000586     0.120467    -0.008669    -0.034222     0.002451
  2     0.000396    -0.005093     0.065502    -0.042772    -0.292349    -0.002298     0.029553
  3     0.037615     0.002873     0.000219     0.183865     0.014061    -0.030572    -0.002335
  4     0.006027    -0.008824     0.012918    -0.000000    -0.018911     0.017026    -0.024926
  5    -0.126410    -0.045356    -0.016274    -0.291746    -0.110518     0.046478     0.016677
STO    -0.000000     0.000000    -0.000000    -0.000000     0.000000     0.000000    -0.000000
  7    -0.227787     0.112367    -0.055431    -0.252955     0.152127     0.038302    -0.018894
  8     0.000130     0.000581     0.002588     0.000000     0.011535     0.003927     0.017502
  9     0.093297    -0.028806     0.008894     0.178746    -0.057936    -0.020104     0.006207
 10    -0.000002    -0.000264    -0.030245     0.033653     0.389861    -0.000233    -0.026621
 11     0.128073    -0.023063     0.004153     0.098441    -0.018475    -0.019011     0.003423
IMA     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000     0.000000
TOT     0.025514    -0.003761    -0.007089     0.027699     0.060727    -0.000706     0.003038

Seidel Aberration Coefficients in Waves:

Surf      W040         W131         W222        W220P         W311         W020         W111
  1    24.288415    -6.957359     0.498230    51.254016    -7.376475   -29.120000     4.170677
  2     0.084231    -4.333486    55.736947   -18.197852  -248.765515    -1.955190    50.294990
  3     8.001724     2.444808     0.186744    78.227152    11.964837   -26.014187    -3.974125
  4     1.282189    -7.508364    10.992049    -0.000000   -16.092073    14.487712   -42.419262
  5   -26.891256   -38.594487   -13.847758  -124.126173   -94.041899    39.549358    28.380735
STO    -0.000000     0.000000    -0.000000    -0.000000     0.000000     0.000000    -0.000000
  7   -48.457040    95.615506   -47.167166  -107.622229   129.447755    32.591819   -32.155114
  8     0.027719     0.494159     2.202392     0.000000     9.815724     3.341623    29.786205
  9    19.847124   -24.511996     7.568325    76.049166   -49.298685   -17.107262    10.564078
 10    -0.000492    -0.224984   -25.736060    14.318059   331.740106    -0.198025   -45.304611
```

Wavelength The wavelength number to use for the calculation.

Discussion

Zemax will compute the unconverted Seidel, transverse, longitudinal, and some wavefront coefficients. The Seidel coefficients are listed surface by surface, as well as a sum for the entire system. The coefficients listed are for spherical aberration (SPHA, S1), coma (COMA, S2), astigmatism (ASTI, S3), field curvature (FCUR, S4), distortion (DIST, S5), longitudinal color (CLA, CL), and transverse color (CTR, CT). The units are always the same as the system lens units, except of course for the coefficients measured in waves.

These calculations are only valid and accurate for systems consisting entirely of surfaces that are axial symmetric spheres, conic, second order, or fourth order aspheres. Any systems which contain coordinate breaks, gratings, paraxial, or other surfaces that are not radially symmetric are not adequately described by the paraxial rays which are used to compute the coefficients. Specifically, the supported surface types are the Standard, Even Asphere, Odd Asphere, Extended Asphere, and Extended Odd Asphere.

Transverse aberration coefficients are listed for each surface and for the system as a whole. The coefficients given are transverse spherical (TSPH), transverse sagittal coma (TSCO), transverse tangential coma (TTCO), transverse astigmatism (TAST), transverse Petzval field curvature (TPFC), transverse sagittal field curvature (TSFC), transverse tangential field curvature (TTFC), transverse distortion (TDIS), transverse axial color (TAXC), and transverse lateral color (TLAC). The transverse aberrations are in the system lens units. The transverse aberration coefficients may be very large in optical spaces where the light is nearly collimated, and have little meaning in these optical spaces.

Longitudinal aberration coefficients are computed for longitudinal spherical aberration (LSPH), longitudinal sagittal coma (LSCO), longitudinal tangential coma (LTCO), longitudinal astigmatism (LAST), longitudinal Petzval field curvature (LPFC), longitudinal sagittal field curvature (LSFC), longitudinal tangential field curvature (LTFC), longitudinal distortion (LDIS), longitudinal axial color (LAXC), and longitudinal lateral color (LLAC). The longitudinal aberrations are in the system lens units. The longitudinal aberration coefficients may be very large in optical spaces where the light is nearly collimated, and have little meaning in these optical spaces.

The wavefront coefficients given are spherical aberration (W040), coma (W131), astigmatism (W222), field curvature Petzval (W220P), distortion (W311), axial color defocus term (W020), lateral color tilt term (W111), field curvature sagittal (W220S), field curvature medial (W220M), and field curvature tangential (W220T). All the wavefront coefficients are in units of wavelengths at the edge of the exit pupil. The various aberration coefficients are interrelated according to the following table. The n' and u' values are the index of refraction and the paraxial marginal ray angle at the last surface in the system. For a discussion of the meaning and derivation of the Seidel aberration coefficients, see Welford, Aberrations of Optical Systems, Smith, Modern Lens Design, or O' Shea, Elements of Modern Optical Design. A list of good references can be found in "REFERENCES ON LENS DESIGN"

Name	Seidel	Wave	Description	Transverse	Longitudinal
Spherical	S_1	$\dfrac{S_1}{8}$	Spherical	$-\dfrac{S_1}{2n'u'}$	$\dfrac{S_1}{2n'u'^2}$
Coma	S_2	$\dfrac{S_2}{2}$	Sagittal	$-\dfrac{S_2}{2n'u'}$	$\dfrac{S_2}{2n'u'^2}$
			Tangential	$-\dfrac{3S_2}{2n'u'}$	$\dfrac{3S_2}{2n'u'^2}$
Astigmatism	S_3	$\dfrac{S_3}{2}$	From tangential to sagittal foci	$-\dfrac{S_3}{n'u'}$	$\dfrac{S_3}{n'u'^2}$
Field Curvature	S_4	$\dfrac{S_4}{4}$	Gaussian to Petzval	$-\dfrac{S_4}{2n'u'}$	$\dfrac{S_4}{2n'u'^2}$
	$S_3 + S_4$	$\dfrac{S_3 + S_4}{4}$	Gaussian to sagittal	$-\dfrac{(S_3 + S_4)}{2n'u'}$	$\dfrac{S_3 + S_4}{2n'u'^2}$
	$2S_3 + S_4$	$\dfrac{2S_3 + S_4}{4}$	Gaussian to medial	$-\dfrac{(2S_3 + S_4)}{2n'u'}$	$\dfrac{2S_3 + S_4}{2n'u'^2}$
	$3S_3 + S_4$	$\dfrac{3S_3 + S_4}{4}$	Gaussian to tangential	$-\dfrac{(3S_3 + S_4)}{2n'u'}$	$\dfrac{3S_3 + S_4}{2n'u'^2}$
Distortion	S_5	$\dfrac{S_5}{2}$	Distortion	$-\dfrac{S_5}{2n'u'}$	$\dfrac{S_5}{2n'u'^2}$
Axial Color	C_L	$\dfrac{C_L}{2}$	Chromatic aberrations are measured between the extreme defined wavelengths, referenced to the selected wavelength.	$\dfrac{C_L}{n'u'}$	$\dfrac{C_L}{n'u'^2}$
Lateral Color	C_T	C_T		$\dfrac{C_T}{n'u'}$	$\dfrac{C_T}{n'u'^2}$

Seidel Diagram

Displays Seidel unconverted aberration coefficients as a bar chart.
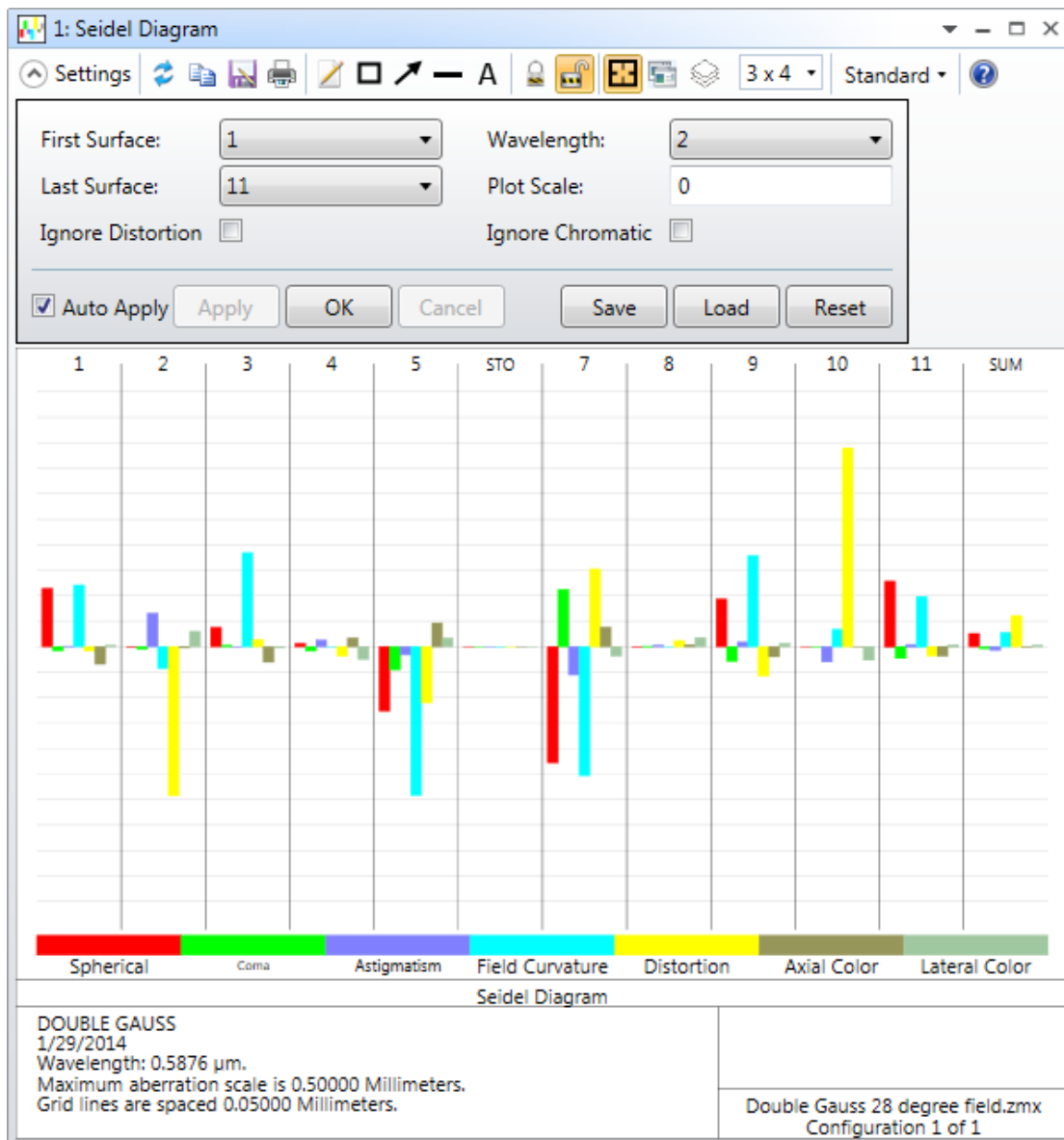
First/Last Surface The surface range to display. The sum is for the selected range of surfaces. The image surface is not displayed.

Wavelength The wavelength number to use for the calculation.

Plot Scale Sets the maximum scale in lens units for the plot. Enter zero for automatic.

Ignore Distortion If checked, distortion data will not be displayed.

Ignore Chromatic If checked, axial and lateral color data will not be displayed.

Discussion The unconverted Seidel coefficients are displayed surface by surface, as well as a sum for the selected range of surfaces. For details on the Seidel coefficients, see "Seidel Coefficients"

Wavefront
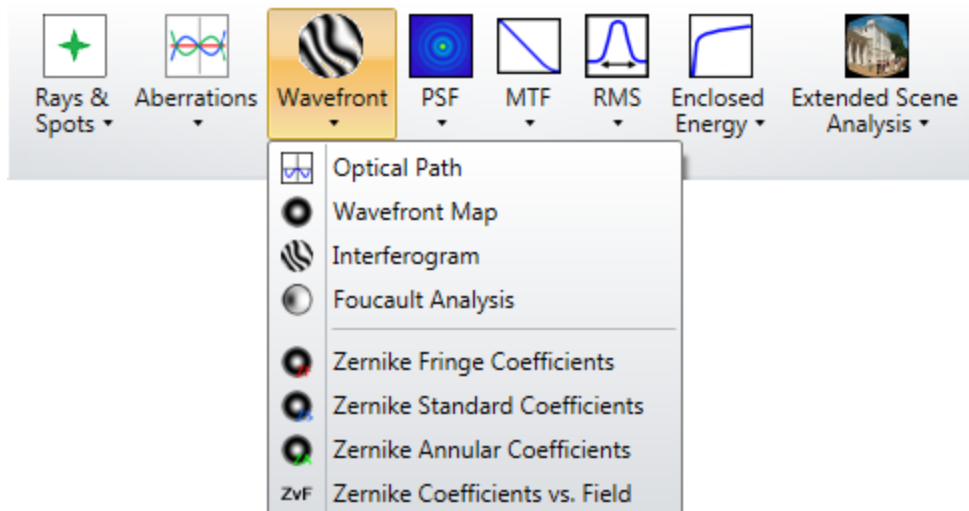
View and analyze the wavefront produced by the optical system

Optical Path

Shows optical path difference as a function of pupil coordinate.

The options are identical to those for ray aberration fans, except the only option for "Tangential" and "Sagittal" is OPD, since OPD is a scalar quantity.

Discussion The vertical axis scale is given at the bottom of the graph. The data being plotted is the optical path difference, or OPD, which is the difference between the optical path length of the ray and the optical path length of the chief ray. Usually, the calculation is referenced back to the difference between the ray path lengths at the system exit pupil. The horizontal scale for each graph is the normalized entrance pupil coordinate.

If "All" wavelengths are shown, then the plot is referenced to the primary wavelength based reference sphere and chief ray. If monochromatic, then the reference sphere and chief ray for the selected wavelength is used as a reference. For this reason, the data for non-primary wavelengths will in general change when switching between monochromatic and polychromatic displays.

Wavefront Map

Displays the wavefront error across the pupil
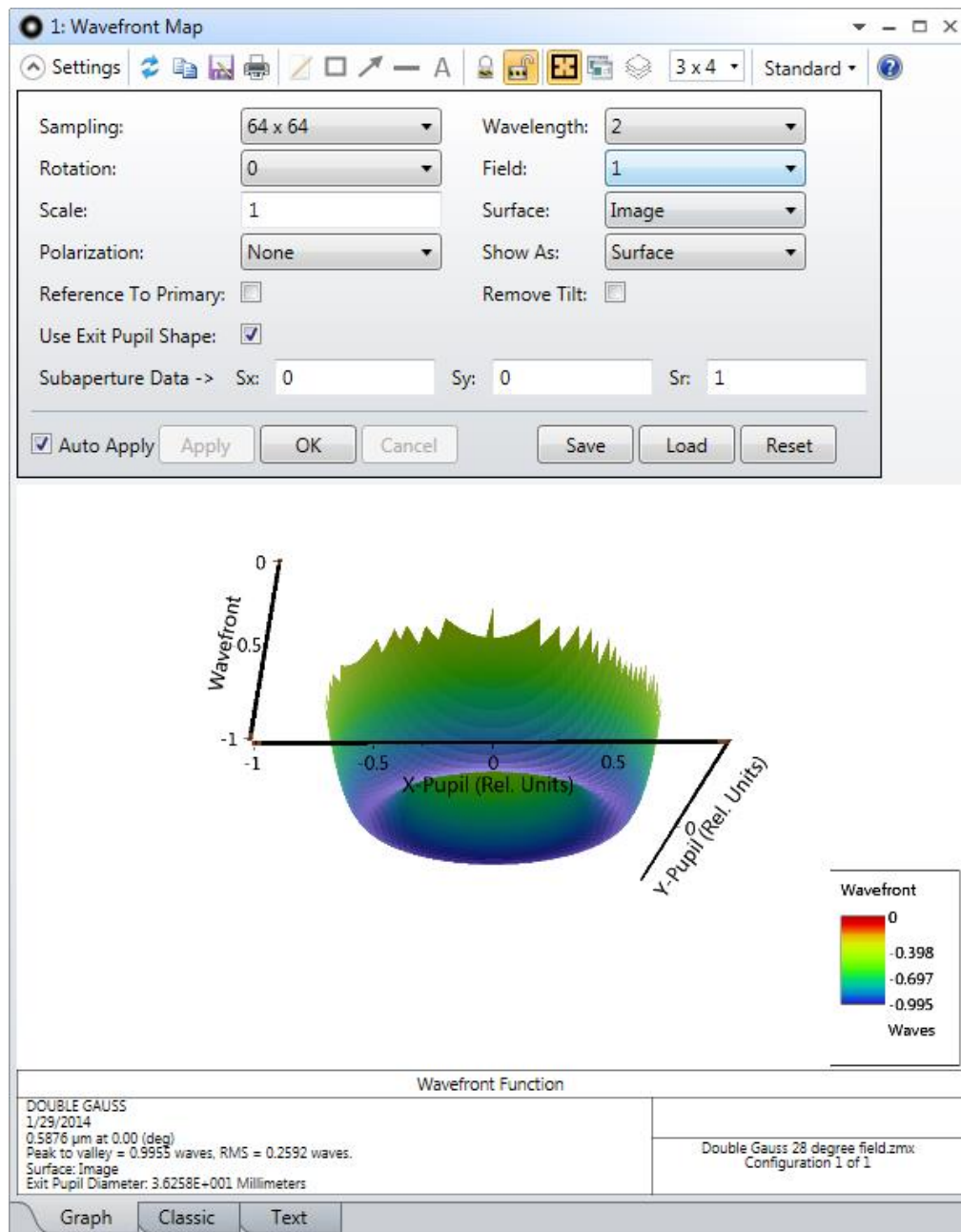
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Rotation Rotation specifies how the surface plots are rotated for viewing; either 0, 90, 180, or 270 degrees.

Scale The scale factor is used to override the automatic vertical scaling set by the program on the surface plots. The scale factor can be greater than unity to vertically stretch the plot, or less than unity to compress it.

Polarization If none is selected, polarization is ignored. If Ex, Ey, or Ez is selected, then the phase due to polarization effects for the specified component of the electric field is added to the optical path difference. If the polarization phase exceeds one wave, the wavefront map may display abrupt changes of 2□ because no phase "unwrapping" is performed.

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Reference To Primary By default, the wavefront aberration is referenced to the reference sphere for the wavelength being used. If this box is checked, then the primary wavelength reference sphere will be used instead. In other words, checking this box will cause the data to exhibit the effects of lateral color.

Use Exit Pupil Shape If checked, the shape of the pupil is distorted to show the approximate shape of the exit pupil as seen from the image point for the specified field. The shape is based upon the F/# of the beam in the X and Y pupil directions. If this box is unchecked, then instead the plot will be scaled to circular entrance pupil coordinates, no matter how distorted the exit pupil may actually be.

Show As Choose surface plot, contour map, grey scale, or false color map as the display option.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Remove Tilt If checked, the linear X- and Y- tilt is removed from the data. This is equivalent to referencing the OPD data to the centroid.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String".

Subaperture Data: Sx, Sy, Sr Defines a subaperture of the pupil to compute the wavefront data for. See "Subaperture computations" for details.

Discussion See also the Interferogram feature. For details on RMS wavefront error see "Comments about

RMS wavefront computations"

Interferogram

Generates and displays interferograms
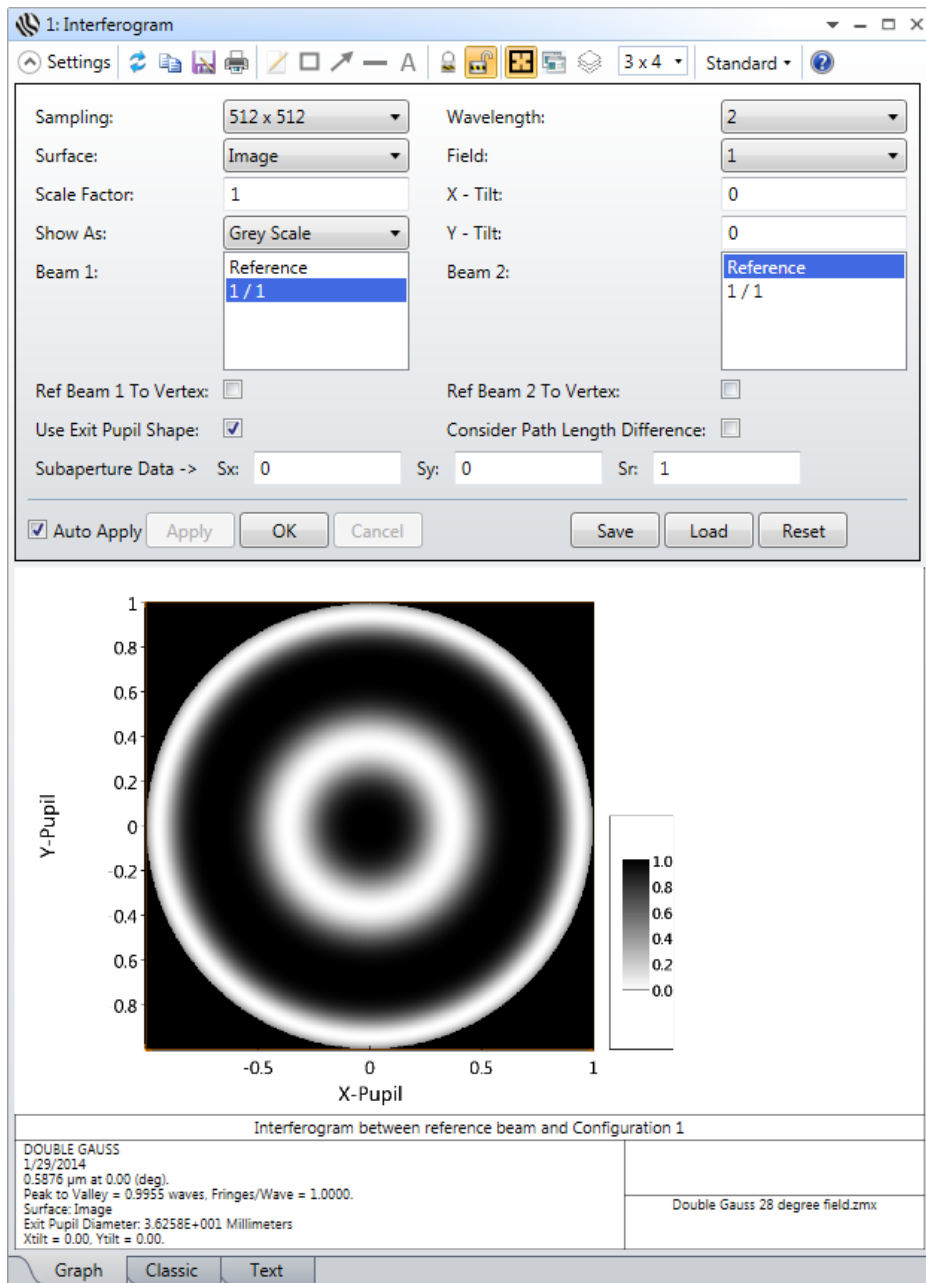
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Scale Factor Determines the number of fringes per wave of OPD. Useful for modeling double pass interferometers (i.e. use a scale factor of two).

Show As Choose contour map, grey scale, or false color map as the display option.

Wavelength The wavelength is the wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

X-Tilt The number of fringes of tilt to add in the x-direction after applying the scale factor.

Y-Tilt The number of fringes of tilt to add in the y-direction after applying the scale factor.

Beam 1 Selects the first beam for the interferogram. See the discussion.

Beam 2 Selects the second beam for the interferogram. See the discussion.

Ref Beam To Vertex Normally, Zemax references the OPD to the chief ray; which in effect subtracts out tilt from the wavefront phase. For interferometry, it is sometimes desirable to retain the wavefront tilt. Checking this option on will add tilt to the beam based upon the deviation of the chief ray from the image surface vertex. This option is only useful for field positions whose chief ray is reasonably close to the surface vertex, where the assumption that tilt is described by the deviation of the chief ray is valid. This feature should also only be used when the scale factor is 1.0.

Use Exit Pupil Shape If checked, the shape of the pupil is distorted to show the approximate shape of the exit pupil as seen from the image point for the specified field. The shape is based upon the F/# of the beam in the X and Y pupil directions. If this box is unchecked, then instead the plot will be scaled to circular entrance pupil coordinates, no matter how distorted the exit pupil may actually be.

If Beam 1 is "reference" then the configuration of Beam 2 is used to determine the pupil shape, otherwise the configuration of Beam 1 is used. If both Beam 1 and Beam 2 are defined by different configurations, the shape of Beam 1 is used, and no attempt is made to confirm that the pupils are the same shape. If the pupil shapes are different for the two configurations, this feature cannot accurately predict the interferogram.

Consider Path Length Difference If unchecked, the interferogram is computed assuming the phase at the center of each beam is zero, resulting in a "dark" fringe at the center. No attempt is made to determine the phases of the two beams with respect to each other.

If checked, the difference between the total optical path length of the chief rays in each beam configuration is considered. This setting will alter the phase of the entire interferogram, but will not change the shape of the fringe pattern. For example, if the path difference is exactly one-half wave, dark fringes will become bright and bright will become dark. The optical path length of the chief rays from object to image surface is used, even if the fringes are not localized at the image surface. This assumption is only valid if the fringes are localized on a plane that is the same distance from the image surfaces in each configuration. If a reference beam is selected rather than a beam from a configuration, the optical path length for the reference beam is assumed to be zero.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String".

Subaperture Data: Sx, Sy, Sr Defines a subaperture of the pupil to compute the interferogram data for. See "Subaperture computations" for details.

Discussion

This feature works by computing two pupil maps, one each from beams 1 and 2. The phase (or OPD) of these two pupil maps is subtracted, and then optionally some linear phase is added as a function of the x and y pupil coordinate to simulate tilt fringes. The individual beams may be OPD as computed for any one configuration, or a "reference" beam which has identically zero OPD may be selected.

Interferometers may be simulated by modeling the two paths through the system using two configurations, and then computing the interferogram of the two resulting beams. The accuracy of this approach is limited by some simplifying assumptions:

- Any lateral shift or magnification difference between the two beams is ignored; it is assumed that the pupils perfectly overlap at the exit pupil.

- Any differences in transmission are ignored; so the two OPD values at any one point in the pupil are assumed to be of equal value in intensity and the phase can be subtracted to yield the net phase difference.

Foucault Analysis

Generates and displays Foucault knife-edge shadowgrams.
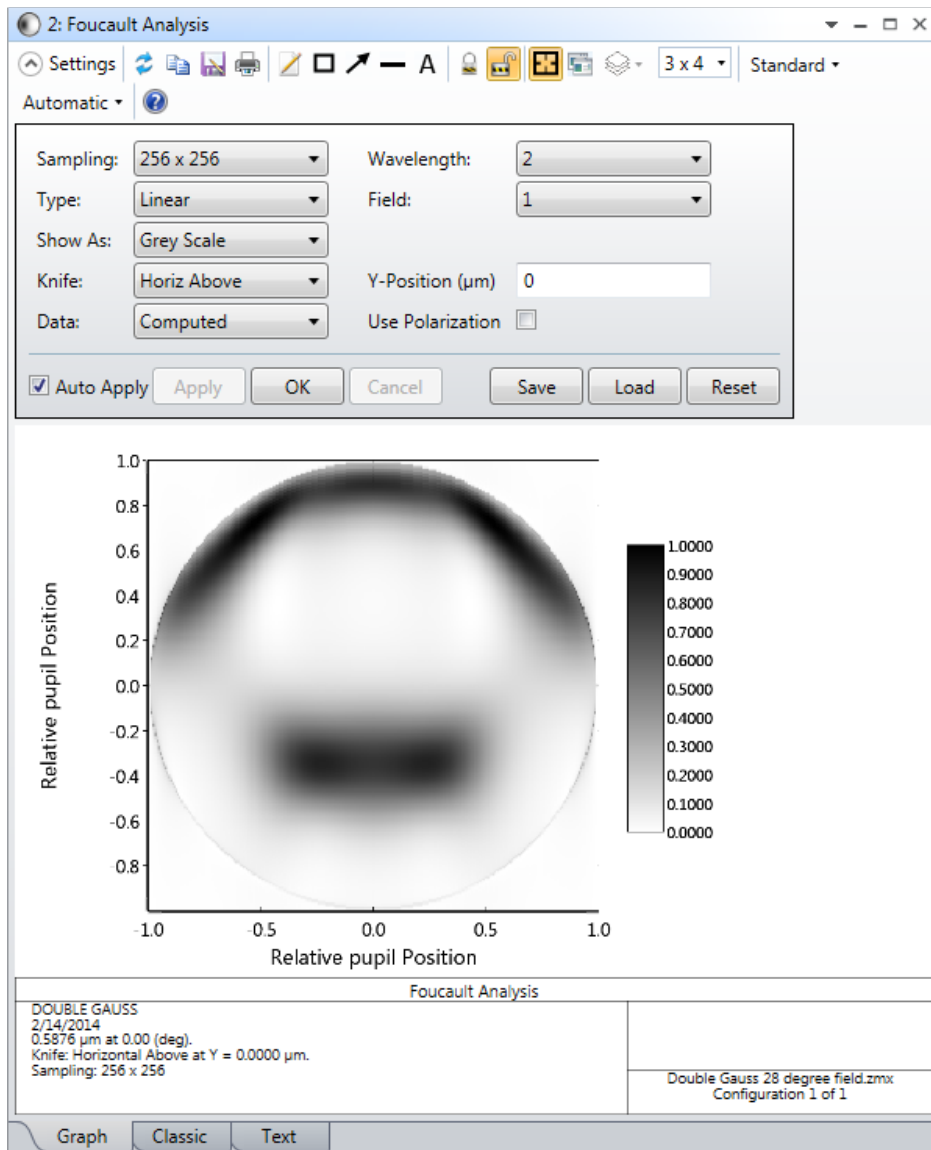
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Type Selects either linear or logarithmic display of the data.

Show As Choose surface, contour map, grey scale, false color map, or cross section as the display option.

Wavelength The wavelength is the wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Row/Column When "Show As" is selected to be a cross section, this control defines the row or column number to display.

Knife Choose Horizontal Above, Horizontal Below, Vertical Left, or Vertical Right. The Vertical Left knife blocks out all the light near focus from the knife position coordinate left; that is, towards negative x coordinates. The Vertical Right blocks all light from the knife position rightward. The Horizontal Above blocks all light from the knife position up, and Horizontal Below blocks all light from the knife position down. The terms left, right, up and down refer to the -x, +x, +y, and -y directions in the local coordinate system of the image surface.

Position The position in micrometers relative to the chief ray of the knife. The coordinate is assumed to be in X or Y depending upon whether an X or Y knife is selected.

Data The computed shadowgram, the reference shadowgram, or the difference between the two may be selected. See the discussion for details.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Reference The name of the bitmap reference image file.

Decenter X/Y The decenter in X or Y of the reference shadowgram image relative to the computed shadowgram image. The units are relative to the full width or height of the reference shadowgram image. For example, an X Decenter of 0.25 will shift the reference image relative to the computed image by 25% of the full width of the reference image.

Scale X/Y The scale factor in X or Y of the reference shadowgram image pixels relative to the computed shadowgram image pixels.

Discussion

This feature simulates the placement of either an X- or Y- oriented knife edge at the image surface; then computes the resulting shadowgram after propagating the vignetted beam back to the near field. The method of calculation involves computing the diffraction based complex amplitude PSF at focus via the FFT method; then a portion of the complex amplitude is vignetted by the simulated knife edge, and the remaining complex amplitude is propagated back to the near field. The shadowgram calculated this way is called the "computed" shadowgram for this feature.

This feature also allows the import of either a bitmap file of a reference or measured shadowgram. The reference shadowgram may be displayed for convenient check of orientation.

The difference between the computed and reference shadowgram may be displayed. Zemax computes the

RMS difference between the computed and reference shadowgrams, and this RMS difference may be optimized

using the FOUC operand described in the chapter "Optimization". Optimizing the RMS difference permits quantitative determination of the aberrations present in the beam that created the measured shadowgram. When calculating the difference between the computed and reference shadowgram, the two images must be registered together to overlap correctly. The decenter x/y and scale x/y controls are used for registering the two images.

Zernike Fringe Coefficients

Calculates the Zernike coefficients using the Fringe (also called the "University of Arizona") polynomials.
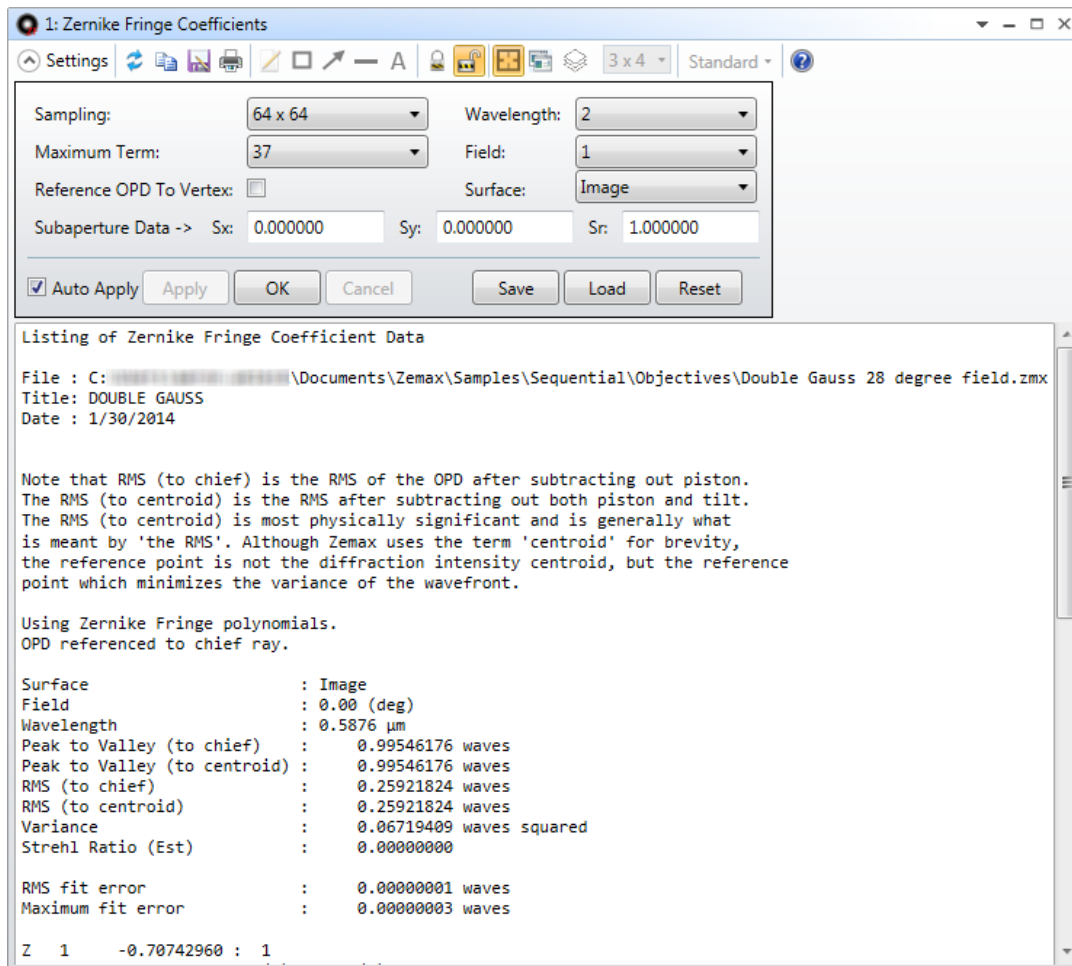
Sampling Specify the density in the pupil to use for coefficient fitting. Larger grid sizes are more accurate, although the computation time increases.

Maximum Term Specify the maximum Zernike coefficient to compute. Any value up to 37 may be specified.

Wavelength The wavelength number to use for the calculation.

Field The field number to use for the calculation.

Ref OPD To Vertex Normally, Zemax references the OPD to the chief ray; which in effect subtracts out tilt from the wavefront phase. For interferometry, it is sometimes desirable to retain the wavefront tilt. Checking this option on will add tilt to the beam based upon the deviation of the chief ray from the image surface vertex. This option is only useful for field positions whose chief ray is reasonably close to the surface vertex, where the assumption that tilt is described by the deviation of the chief ray is valid.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Subaperture Data: Sx, Sy, Sr Defines a subaperture of the pupil to compute the Zernike data for. See "Subaperture computations" for details.

Discussion

This feature displays the individual Zernike coefficients as well as the peak-to-valley, RMS, variance, Strehl ratio, residual RMS fit error, and maximum fit error.

The RMS of the wavefront error, σ , when referenced to the mean, is defined as

$$\sigma^2 = \overline{W^2} - \overline{W}^2$$

where W is the wavefront error, $\overline{W^2}$ is the mean square error, and $\overline{W}$ is the average wavefront error. The RMS can actually be computed several different ways. If the mean wavefront term is ignored, then the RMS "referenced to zero" results. This computation yields the square root of $\overline{W^2}$ directly, and is rarely used.

If the mean wavefront is subtracted from all the wavefront phase values (the absolute phase reference has no physical meaning), then the RMS is "referenced to the mean".

Typically, the RMS is further referenced to the tilted and shifted reference sphere which minimizes the RMS. This is equivalent to subtracting out not only the mean (which is piston) but the average tilt in x and y as well. This is justified because tilt shifts the location of the diffraction image centroid, but otherwise has no effect on image quality. For brevity, Zemax calls this reference point the "centroid", although it is a reference point which is usually close to but not exactly at the diffraction image centroid. Most of the time, the RMS is taken to mean the RMS referenced to the centroid, which is always the lowest of the three numbers.

The W values used by the Zernike RMS computations are the "raw" OPD values measured by computing the phase of each ray as it intercepts the reference sphere. The reference sphere is centered on the chief ray - image surface intercept point, and has a radius equal to the "exit pupil position" length (see "Exit pupil position"). This method does not consider a subtle effect due to the change in the location of the center of the reference sphere in the presence of coma. For this reason, the Zernike based estimate of the RMS wavefront referenced to the centroid may differ slightly from that computed by the RMS analysis feature; see "Comments about RMS wavefront computations" for more information.

Subaperture computations

By default, the Zernike polynomials are determined for the entire pupil. The subaperture values Sx, Sy, and Sr allow computation of the Zernike data only over a circular subaperture portion of the pupil. The subaperture is a circular region of normalized radius Sr, decentered from the usual pupil by the normalized coordinates Sx and Sy.
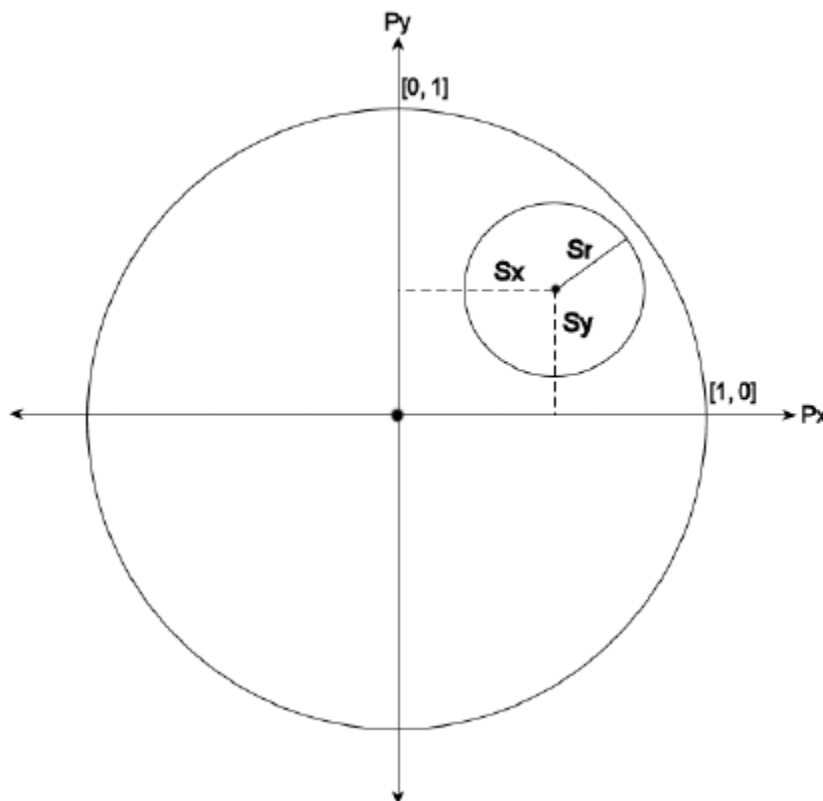
All three subaperture data values (Sx, Sy, and Sr) are normalized dimensionless parameters relative to the full pupil unit circle. The transformation between full aperture and subaperture normalized pupil coordinates is given by:

$$P_x = S_x + S_r \times F_x, \text{ and } P_y = S_y + S_r \times F_y$$

where Fx and Fy denote the subaperture coordinates and Px and Py denote the full aperture coordinates.

Strehl ratio approximation

The Strehl ratio is computed using the RMS referenced to the centroid by the following approximation:

$$S = e^{-(2\pi\sigma)^2}$$

This approximation is valid for monochromatic calculations resulting in a Strehl ratio higher than about 0.10.

This feature computes a maximum of 37 Zernike terms. The particular Zernike terms used are not orthonormal, but are instead all normalized to have unity magnitude at the edge of the pupil. Some of the higher order terms in the expansion were dropped to keep the total number of terms small, and the terms remaining were selected to favor accurate fitting of higher order spherical aberration. This particular set of Zernike polynomials is sometimes called the "Fringe" or "University of Arizona" notation. The more formal, and more general polynomial set is the Standard notation, sometimes called the "Born & Wolf" or the similar "Noll" notation, which is described under the "Zernike Standard Coefficients" feature.

The Zernike Fringe polynomials are defined in the following table. The angle φ is measured counter clockwise from the local +x axis. The radial coordinate is the normalized dimensionless parameter ρ.

ZERNIKE FRINGE POLYNOMIALS

Term	$Z(\rho, \varphi)$
1	1
2	$\rho \cos\varphi$
3	$\rho \sin\varphi$
4	$2\rho^2 - 1$
5	$\rho^2 \cos 2\varphi$
6	$\rho^2 \sin 2\varphi$
7	$(3\rho^2 - 2)\rho \cos\varphi$
8	$(3\rho^2 - 2)\rho \sin\varphi$
9	$6\rho^4 - 6\rho^2 + 1$
10	$\rho^3 \cos 3\varphi$
11	$\rho^3 \sin 3\varphi$
12	$(4\rho^2 - 3)\rho^2 \cos 2\varphi$
13	$(4\rho^2 - 3)\rho^2 \sin 2\varphi$
14	$(10\rho^4 - 12\rho^2 + 3)\rho \cos\varphi$
15	$(10\rho^4 - 12\rho^2 + 3)\rho \sin\varphi$
16	$20\rho^6 - 30\rho^4 + 12\rho^2 - 1$
17	$\rho^4 \cos 4\varphi$
18	$\rho^4 \sin 4\varphi$
19	$(5\rho^2 - 4)\rho^3 \cos 3\varphi$
20	$(5\rho^2 - 4)\rho^3 \sin 3\varphi$
21	$(15\rho^4 - 20\rho^2 + 6)\rho^2 \cos 2\varphi$

Term	$Z(\rho, \varphi)$
22	$(15\rho^4 - 20\rho^2 + 6)\rho^2 \sin 2\varphi$
23	$(35\rho^6 - 60\rho^4 + 30\rho^2 - 4)\rho \cos \varphi$
24	$(35\rho^6 - 60\rho^4 + 30\rho^2 - 4)\rho \sin \varphi$
25	$70\rho^8 - 140\rho^6 + 90\rho^4 - 20\rho^2 + 1$
26	$\rho^5 \cos 5\varphi$
27	$\rho^5 \sin 5\varphi$
28	$(6\rho^2 - 5)\rho^4 \cos 4\varphi$
29	$(6\rho^2 - 5)\rho^4 \sin 4\varphi$
30	$(21\rho^4 - 30\rho^2 + 10)\rho^3 \cos 3\varphi$
31	$(21\rho^4 - 30\rho^2 + 10)\rho^3 \sin 3\varphi$
32	$(56\rho^6 - 105\rho^4 + 60\rho^2 - 10)\rho^2 \cos 2\varphi$
33	$(56\rho^6 - 105\rho^4 + 60\rho^2 - 10)\rho^2 \sin 2\varphi$
34	$(126\rho^8 - 280\rho^6 + 210\rho^4 - 60\rho^2 + 5)\rho \cos \varphi$
35	$(126\rho^8 - 280\rho^6 + 210\rho^4 - 60\rho^2 + 5)\rho \sin \varphi$
36	$252\rho^{10} - 630\rho^8 + 560\rho^6 - 210\rho^4 + 30\rho^2 - 1$
37	$924\rho^{12} - 2772\rho^{10} + 3150\rho^8 - 1680\rho^6 + 420\rho^4 - 42\rho^2 + 1$

Zernike Standard Coefficients

Calculates the orthonormal Zernike coefficients computed using the notation defined in R. Noll, "Zernike polynomials and atmospheric turbulence", J. Opt. Soc. Am., Vol. 66, No. 3, p207 (1976). This notation allows terms to be consistently defined up to (almost) arbitrary order, rather than the limited 37 term Fringe polynomials.

Sampling Specify the density in the pupil to use for coefficient fitting. Larger grid sizes are more accurate, although the computation time increases.

Maximum Term Specify the maximum Zernike coefficient to compute. Any value up to 231 may be specified.

Wavelength The wavelength number to use for the calculation.

Field The field number to use for the calculation.

Ref OPD To Vertex Normally, Zemax references the OPD to the chief ray; which in effect subtracts out tilt from the wavefront phase. For interferometry, it is sometimes desirable to retain the wavefront tilt. Checking this option on will add tilt to the beam based upon the deviation of the chief ray from the image surface vertex. This option is only useful for field positions whose chief ray is reasonably close to the surface vertex, where the assumption that tilt is described by the deviation of the chief ray is valid.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Subaperture Data: Sx, Sy, Sr Defines a subaperture of the pupil to compute the Zernike data for. See "Subaperture computations" for details.

Discussion

This feature is similar to the "Zernike Fringe Coefficients" feature, with a somewhat different numbering scheme is used, more terms in the expansion are available for fitting, the terms are orthogonal and normalized, and none of the terms are skipped. See the description of "Zernike Fringe Coefficients" for some important details.

The most general way to express the Zernike polynomials is in the form

$$R_n^m(\rho)e^{im\theta} = \begin{cases} R_n^m(\rho)\cos m\theta \\ R_n^m(\rho)\sin m\theta \end{cases}$$

The radial portion of the polynomial is defined by two indices, n and m. The n index defines the order of the radial power; so an n value of 5 would indicate all polynomials whose maximum radial power was □5 . Only certain values for m are allowed once n is chosen; n + m must be even, and $0 \le m \le n$. For details on Zernike Standard polynomials, see "Principles of Optics", by Born & Wolf, referenced in Chapter 1, or the Noll reference listed above. The terms are orthonormal such that the magnitude of the coefficient of each term is the RMS contribution of the term.

This feature lists the formulas next to the fitted coefficients; the entire 231 term table is too long to include here. The first 28 terms are given below. The angle φ is measured counter clockwise from the local +x axis. The radial coordinate is the normalized dimensionless parameter ρ.

ZERNIKE STANDARD POLYNOMIALS

Term	$Z(\rho, \varphi)$
1	1
2	$\sqrt{4}\,\rho\cos\varphi$
3	$\sqrt{4}\,\rho\sin\varphi$
4	$\sqrt{3}\,(2\rho^2 - 1)$
5	$\sqrt{6}\,(\rho^2\,\sin 2\varphi)$
6	$\sqrt{6}\,(\rho^2\,\cos 2\varphi)$
7	$\sqrt{8}\,(3\rho^3 - 2\rho)\sin\varphi$
8	$\sqrt{8}\,(3\rho^3 - 2\rho)\cos\varphi$
9	$\sqrt{8}\,\rho^3\sin 3\varphi$
10	$\sqrt{8}\,\rho^3\cos 3\varphi$
11	$\sqrt{5}\,(6\rho^4 - 6\rho^2 + 1)$
12	$\sqrt{10}\,(4\rho^4 - 3\rho^2)\cos 2\varphi$
13	$\sqrt{10}\,(4\rho^4 - 3\rho^2)\sin 2\varphi$
14	$\sqrt{10}\,\rho^4\cos 4\varphi$
15	$\sqrt{10}\,\rho^4\sin 4\varphi$
16	$\sqrt{12}\,(10\rho^5 - 12\rho^3 + 3\rho)\cos\varphi$
17	$\sqrt{12}\,(10\rho^5 - 12\rho^3 + 3\rho)\sin\varphi$
18	$\sqrt{12}\,(5\rho^5 - 4\rho^3)\cos 3\varphi$
19	$\sqrt{12}\,(5\rho^5 - 4\rho^3)\sin 3\varphi$

20	$\sqrt{12}\,\rho^5 \cos 5\varphi$
21	$\sqrt{12}\,\rho^5 \sin 5\varphi$
22	$\sqrt{7}\,(20\rho^6 - 30\rho^4 + 12\rho^2 - 1)$
23	$\sqrt{14}\,(15\rho^6 - 20\rho^4 + 6\rho^2)\sin 2\varphi$
24	$\sqrt{14}\,(15\rho^6 - 20\rho^4 + 6\rho^2)\cos 2\varphi$
25	$\sqrt{14}\,(6\rho^6 - 5\rho^4)\sin 4\varphi$
26	$\sqrt{14}\,(6\rho^6 - 5\rho^4)\cos 4\varphi$
27	$\sqrt{14}\,\rho^6 \sin 6\varphi$
28	$\sqrt{14}\,\rho^6 \cos 6\varphi$

Zernike Annular Coefficients

Calculates the orthonormal Zernike coefficients computed using the notation defined in R. Noll, "Zernike polynomials and atmospheric turbulence", J. Opt. Soc. Am., Vol. 66, No. 3, p207 (1976), with extensions for annular pupils, as described in V. N. Mahajan, "Zernike annular polynomials for imaging systems with annular pupils", J. Opt. Soc. Am., Vol. 71, No. 1, p75 (1981). Thanks are due to Brett Patterson of the UK Astronomy Technology Centre who assisted in the development of this feature.

Sampling Specify the density in the pupil to use for coefficient fitting. Larger grid sizes are more accurate, although the computation time increases.

Maximum Term Specify the maximum Zernike coefficient to compute. Any value up to 231 may be specified.

Wavelength The wavelength number to use for the calculation.

Field The field number to use for the calculation.

Ref OPD To Vertex Normally, Zemax references the OPD to the chief ray; which in effect subtracts out tilt from the wavefront phase. For interferometry, it is sometimes desirable to retain the wavefront tilt. Checking this option on will add tilt to the beam based upon the deviation of the chief ray from the image surface vertex. This option is only useful for field positions whose chief ray is reasonably close to the surface vertex, where the assumption that tilt is described by the deviation of the chief ray is valid.

Obscuration The obscuration ratio, ε of the annular pupil.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Subaperture Data: Sx, Sy, Sr Defines a subaperture of the pupil to compute the Zernike data for. See "Subaperture computations" for details.

Discussion

This feature is similar to the "Zernike Standard Coefficients" feature, except the pupil may be annular rather than circular. See that discussion, and the references above, for important information. The annular nature of the pupil is defined by the single parameter ε, defined as

$$\varepsilon = \frac{R_{inner}}{R_{outer}}$$

where R_{inner} and R_{outer} are the inner and outer radii of the annular pupil. When ε is zero, the annulus is a circle, and the Annular and Standard Zernike polynomials are identical. When ε is greater than zero, the annular Zernike polynomials are distinct from the Standard Zernike polynomials. Zemax only traces rays in the annular region when fitting annular polynomials, and unvignetted wavefront outside the annular region is ignored.

The annular polynomials are too complex in general to list on the text output window, so the first 22 terms are listed here. The angle φ is measured counter clockwise from the local +x axis. The radial coordinate is the normalized dimensionless parameter ρ.

ZERNIKE ANNULAR POLYNOMIALS

Term	$Z(\rho, \varphi, \epsilon)$
1	1
2	$\dfrac{2\rho\,\cos\varphi}{\sqrt{1+\epsilon^2}}$
3	$\dfrac{2\rho\,\sin\varphi}{\sqrt{1+\epsilon^2}}$
4	$\dfrac{\sqrt{3}\,(1+\epsilon^2-2\rho^2\,)}{\epsilon^2-1}$
5	$\dfrac{\sqrt{6}\,\rho^2\,\sin 2\varphi}{\sqrt{1+\epsilon^2+\epsilon^4}}$
6	$\dfrac{\sqrt{6}\,\rho^2\,\cos 2\varphi}{\sqrt{1+\epsilon^2+\epsilon^4}}$
7	$\dfrac{2\sqrt{2}\,\rho\,(-2-2\epsilon^4+3\rho^2+\epsilon^2(-2+3\rho^2))\sin\varphi}{\sqrt{(\epsilon^2-1)^2(1+\epsilon^2)(1+4\epsilon^2+\epsilon^4)}}$
8	$\dfrac{2\sqrt{2}\,\rho^3(-2-2\epsilon^4+3\rho^2+\epsilon^2(-2+3\rho^2))\cos\varphi}{\sqrt{(\epsilon^2-1)^2(1+\epsilon^2)(1+4\epsilon^2+\epsilon^4)}}$
9	$\dfrac{2\sqrt{2}\,\rho^3\sin 3\varphi}{\sqrt{1+\epsilon^2+\epsilon^4+\epsilon^6}}$
10	$\dfrac{2\sqrt{2}\,\rho^3\cos 3\varphi}{\sqrt{1+\epsilon^2+\epsilon^4+\epsilon^6}}$

11	$$\frac{\sqrt{5}\,(1+\epsilon^4-6\rho^2+6\rho^4+\epsilon^2(4-6\rho^2))}{(\epsilon^2-1)^2}$$
12	$$\frac{\sqrt{10}\,\sqrt{\frac{(\epsilon^2-1)^4\,(1+\epsilon^2+\epsilon^4)}{1+4\epsilon^2+10\epsilon^4+4\epsilon^6+\epsilon^8}}\,\rho^2\big(3+3\epsilon^6-4\rho^2+\epsilon^2(3-4\rho^2)+\epsilon^4(3-4\rho^2)\big)\cos 2\varphi}{(\epsilon^2-1)^3(1+\epsilon^2+\epsilon^4)}$$
13	$$\frac{\sqrt{10}\,\sqrt{\frac{(\epsilon^2-1)^4\,(1+\epsilon^2+\epsilon^4)}{1+4\epsilon^2+10\epsilon^4+4\epsilon^6+\epsilon^8}}\,\rho^2\big(3+3\epsilon^6-4\rho^2+\epsilon^2(3-4\rho^2)+\epsilon^4(3-4\rho^2)\big)\sin 2\varphi}{(\epsilon^2-1)^3(1+\epsilon^2+\epsilon^4)}$$
14	$$\frac{\sqrt{10}\,\rho^4\cos 4\varphi}{\sqrt{1+\epsilon^2+\epsilon^4+\epsilon^6+\epsilon^8}}$$
15	$$\frac{\sqrt{10}\,\rho^4\sin 4\varphi}{\sqrt{1+\epsilon^2+\epsilon^4+\epsilon^6+\epsilon^8}}$$
16	$$\frac{-2\sqrt{3}\,(E)(Q)\,\rho\cos\varphi}{(\epsilon^2-1)^3(1+4\epsilon^2+\epsilon^4)}\ ,\text{ where}$$ $$E = \sqrt{\frac{(\epsilon^2-1)^2(1+4\epsilon^2+\epsilon^4)}{1+9\epsilon^2+9\epsilon^4+\epsilon^6}}\ ,\text{ and}$$ $$Q = 3+3\epsilon^8-12\rho^2+10\rho^4-12\epsilon^6(\rho^2-1)+2\epsilon^4(15-24\rho^2+5\rho^4)+4\epsilon^2(3-12\rho^2+10\rho^4)\ .$$
17	$$\frac{-2\sqrt{3}\,(E)(Q)\,\rho\sin\varphi}{(\epsilon^2-1)^3(1+4\epsilon^2+\epsilon^4)}\ ,\text{ where}$$ E and Q are defined in term 16 above.
18	$$\frac{-2\sqrt{3}\,(E)(Q)\,\rho^3\cos 3\varphi}{(\epsilon^2-1)^4(1+\epsilon^2)(1+\epsilon^4)}\ ,\text{ where}$$ $$E = \sqrt{\frac{(\epsilon^2-1)^6(1+\epsilon^2)(1+\epsilon^4)}{1+4\epsilon^2+10\epsilon^4+20\epsilon^6+10\epsilon^8+4\epsilon^{10}+\epsilon^{12}}}\ ,\text{ and}$$ $$Q = 4+4\epsilon^8-5\rho^2+\epsilon^2(4-5\rho^2)+\epsilon^4(4-5\rho^2)+\epsilon^6(4-5\rho^2)\ .$$
19	$$\frac{-2\sqrt{3}\,(E)(Q)\,\rho^3\sin 3\varphi}{(\epsilon^2-1)^4(1+\epsilon^2)(1+\epsilon^4)}\ ,\text{ where}$$ E and Q are defined in term 18 above.
20	$$\frac{3\sqrt{3}\,\rho^5\cos 5\varphi}{\sqrt{1+\epsilon^2+\epsilon^4+\epsilon^6+\epsilon^8+\epsilon^{10}}}$$
21	$$\frac{3\sqrt{3}\,\rho^5\sin 5\varphi}{\sqrt{1+\epsilon^2+\epsilon^4+\epsilon^6+\epsilon^8+\epsilon^{10}}}$$
22	$$\frac{\sqrt{7}\,(1+\epsilon^6-12\rho^2+30\rho^4-20\rho^6-3\epsilon^4(-3+4\rho^2)+3\epsilon^2(3-12\rho^2+10\rho^4))}{(\epsilon^2-1)^3}$$

Zernike Coefficients vs. Field

Calculates the Fringe, Standard or Annular Zernike coefficients across the field at the image surface and plots the results.
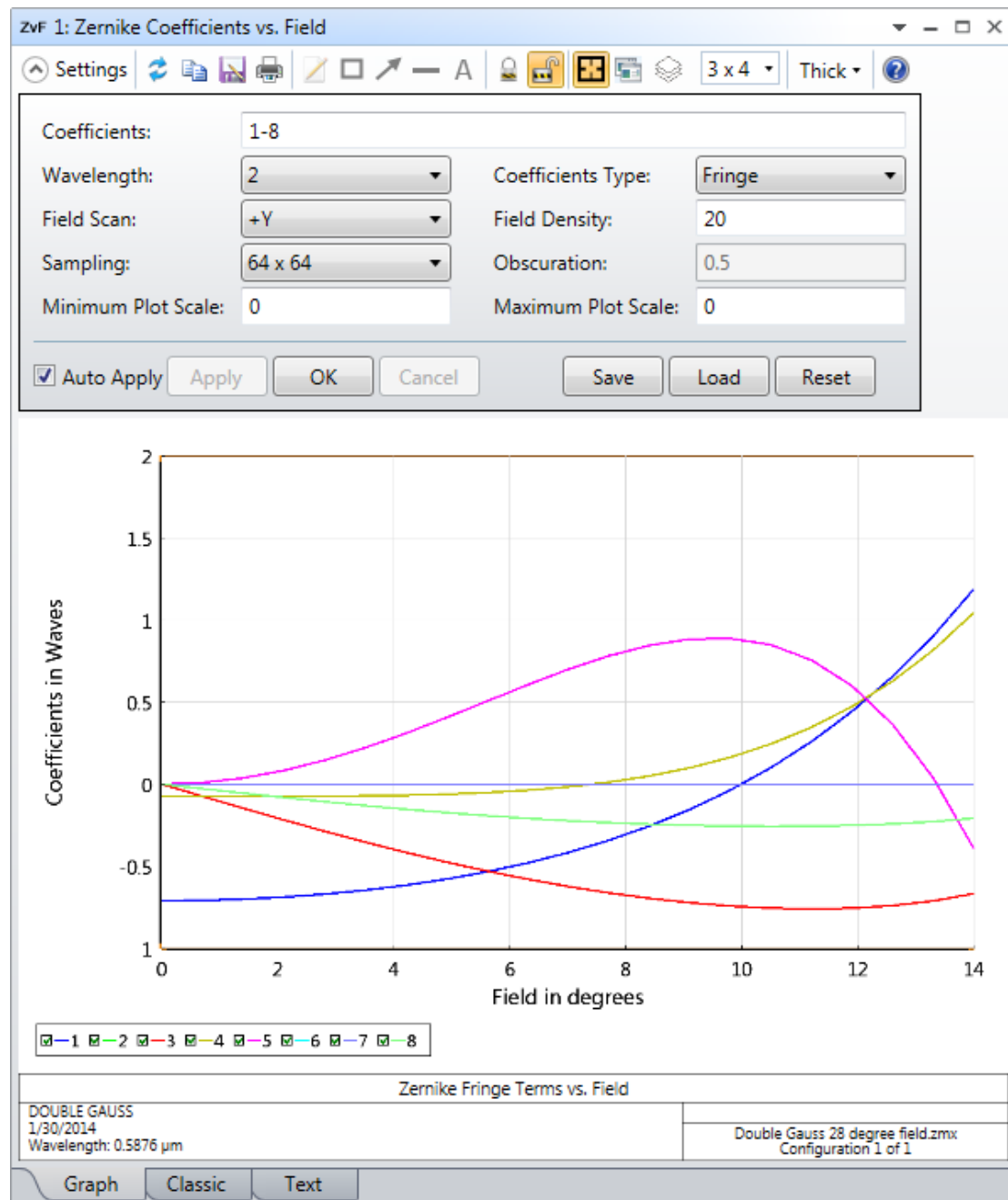
Coefficients A space or comma separated, monotonically increasing, list of integers that corresponds to the terms in the Fringe, Standard or Annular Zernike polynomial. For ranges of terms, separate the first and last term in the range by a dash. Note that when the Coefficients Type option is set to Fringe, the maximum number of terms is 37; if it is set to Standard or Annular then 231 terms are permitted. If zero is used then all coefficients are plotted.

Wavelength The wavelength number to use for the calculation.

Coefficients Type Choose to plot the Fringe, Standard, or Annular Zernike polynomials.

Field Scan Choose to calculate the Zernike terms along the positive X or Y, or negative X or Y field directions.

Field Density The number of field points used to calculate each Zernike term between the maximum and minimum field defined in the Field dialog.

Sampling The density in the pupil to use for coefficient fitting. Larger grid sizes are more accurate, although the computation time increases.

Obscuration The obscuration ratio of the annular pupil.

Minimum/Maximum Plot Scale The range of values displayed on the y-axis. Zero indicates Zemax will choose default values.

Discussion

The Zernike Fringe, Standard or Annular coefficients are displayed as a function of the +y, -y, +x, or -x field. Specific terms or ranges of terms can be selected for display by using a comma or space delimited list of integers. Ranges are defined by separating the first and last term to be displayed by a dash. For example, the list 1, 5-7 would plot the Zernike terms 1 and 5 through 7. For a complete set of Fringe coefficients see the "Zernike Fringe Coefficients". For a complete set of Standard coefficients see the "Zernike Standard Coefficients". It should be noted that the effect of vignetting factors on the calculation is ignored. The obscuration ratio is described in full in the "Zernike Annular Coefficients"

PSF
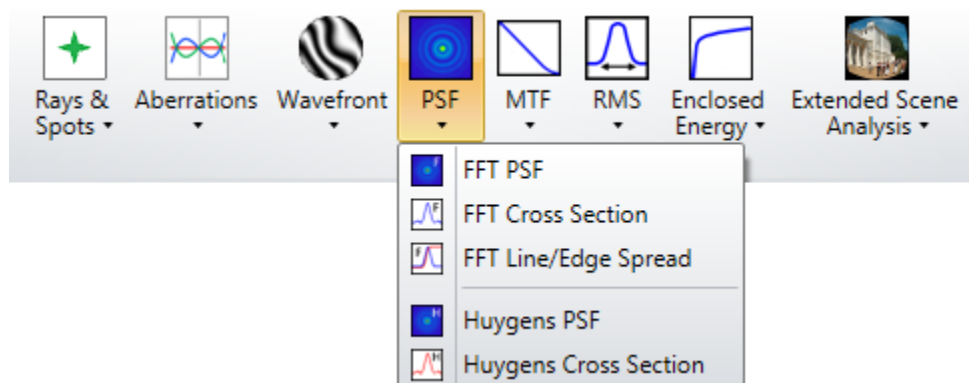
Compute the Point Spread Function (PSF) of a lens using various algorithms

FFT PSF

Computes the diffraction point spread function (PSF) using the Fast Fourier Transform (FFT) method.
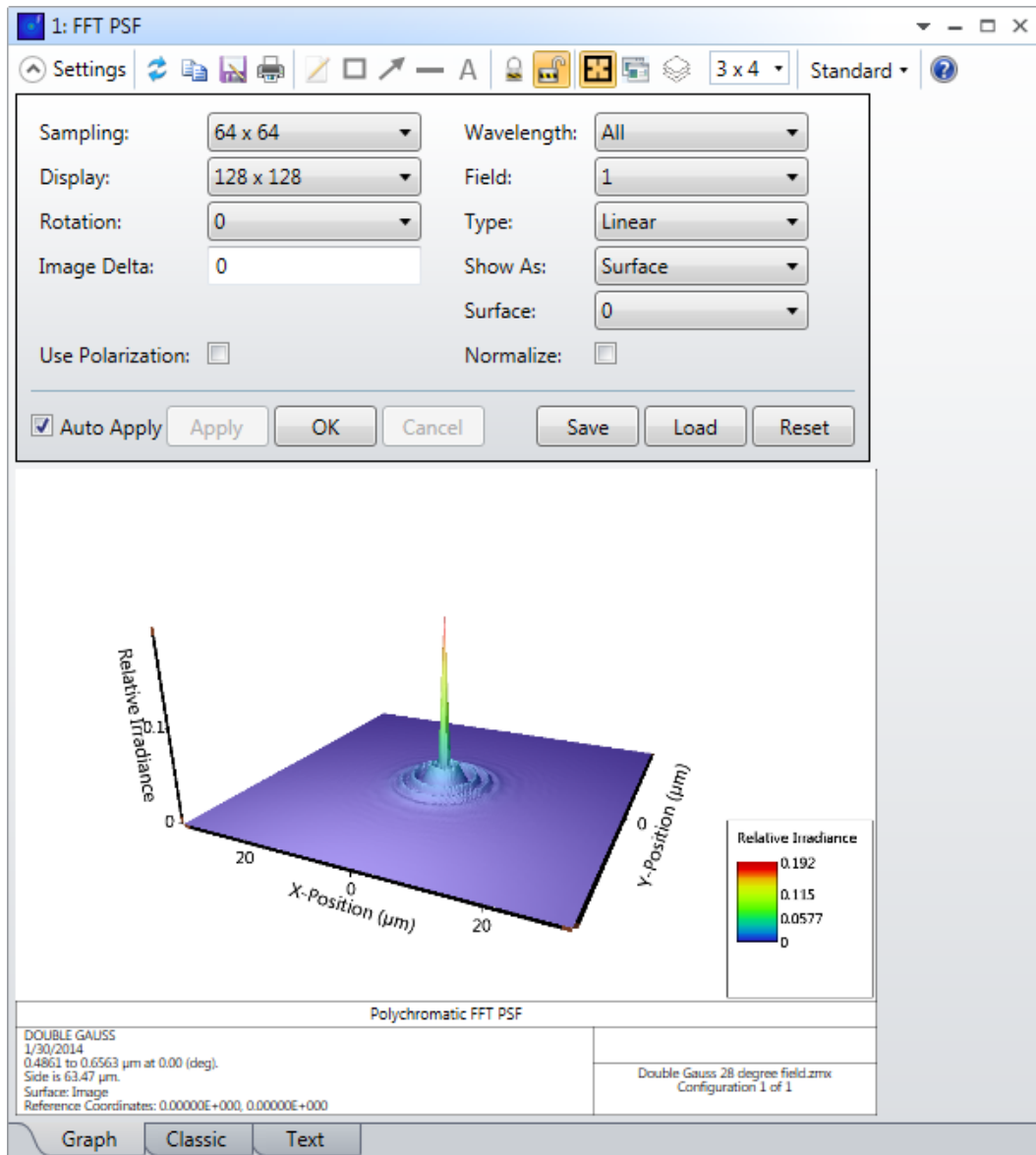
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Display The display size indicates what portion of the computed data will be drawn when a graphic display is generated. The display grid can be any size from 32 x 32 up to twice the sampling grid size. Smaller display sizes will show less data, but at higher magnification for better visibility. This control has no effect on the text display.

Rotation Rotation specifies how the surface plots are rotated for viewing; either 0, 90, 180, or 270 degrees.

Wavelength The wavelength number to be used in the calculation.

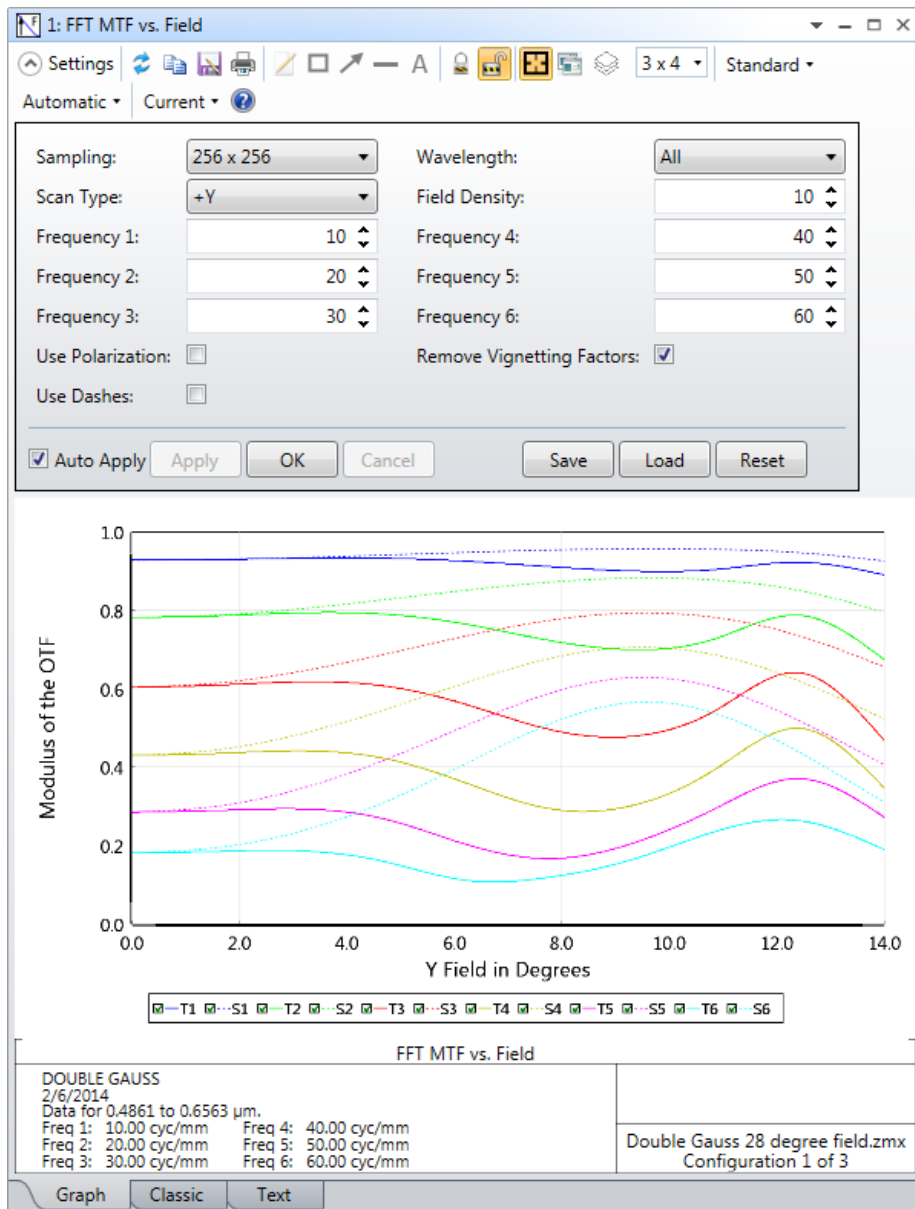**Field** The field number for which the calculation should be performed.

Type Select linear (intensity), logarithmic (intensity), phase, real part (signed amplitude), or imaginary part (signed amplitude).

Show As Choose surface plot, contour map, grey scale, or false color map as the display option.

Use Polarization if checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Image Delta The delta distance between points in image space, measured in micrometers. If zero, a default spacing is used. If negative, the image delta is set to the maximum allowed value and the full sampling grid is used. See the discussion for details.

Normalize If checked, the peak intensity will be normalized to unity. Otherwise, the peak intensity is normalized to the peak of the unaberrated PSF (the Strehl ratio).

Surface Selects the surface at which the PSF is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Discussion

The FFT method of computing the PSF is very fast, however, a few assumptions are made which are not always valid. The slower, but more general Huygens method makes fewer assumptions, and is described in "Huygens PSF"

Assumptions used in the FFT PSF calculation

The FFT PSF computes the intensity of the diffraction image formed by the optical system for a single point source in the field. The intensity is computed on an imaginary plane which is centered on and lies perpendicular to the incident chief ray at the reference wavelength. The reference wavelength is the primary wavelength for polychromatic computations, or the wavelength being used for monochromatic calculations. Because the imaginary plane lies normal to the chief ray, and not the image surface, the FFT PSF computes overly optimistic (a smaller PSF) results when the chief ray angle of incidence is not zero. This is often the case for systems with tilted image surfaces, wide angle systems, systems with aberrated exit pupils, or systems far from the telecentric condition.

The other main assumption the FFT method makes is that the image surface lies in the far field of the optical beam. This means the computed PSF is only accurate if the image surface is fairly close to the geometric focus for all rays; or put another way, that the transverse ray aberrations are not too large. There is no hard and fast limit, however if the transverse aberrations exceed a few hundred wavelengths, the computation is likely not accurate. Note that even systems with very little wavefront aberration can have large transverse ray aberrations; for example, a cylinder lens which only focuses rays along one direction. In this case, the transverse aberrations along the unfocused direction will be on the order of the beam diameter. The Huygens PSF method may provide more accurate results in these cases as well.

For most lenses, a less important assumption is that scalar diffraction theory applies. The vectorial nature of the light is not accounted for. This is significant in systems that are very fast, around F/1.5 (in air) or faster. The scalar theory predicts overly optimistic (a smaller PSF) results when the F/# is very fast.

For systems where the chief ray is nearly normal (less than perhaps 20 degrees), the exit pupil aberrations are negligible, and the transverse ray aberrations are reasonable, then the FFT PSF is accurate and generally much faster than the Huygens PSF method.

When in doubt, both PSF methods should be employed for comparison. A solid understanding on the part of the user of these assumptions and the method of computation is essential to recognize cases where the accuracy may be compromised.

Discussion of the FFT method and sampling issues

The FFT PSF algorithm exploits the fact that the diffraction PSF is related to the Fourier transform of the complex amplitude of the wavefront in the exit pupil of the optical system. The amplitude and phase in the exit pupil are computed for a grid of rays, an FFT is performed, and the diffraction image intensity is computed.

There is a tradeoff between the sampling grid size in the pupil, and the sampling period in the diffraction image. For example, to decrease the sampling period in the diffraction image, the sampling period in the pupil must increase. This is done by "stretching" the pupil sampling grid so that it overfills the pupil. This process means fewer points actually lie within the pupil.

As the sampling grid size is increased, Zemax scales the grid on the pupil to yield an increase in the number of points that lie on the pupil, while simultaneously yielding closer sampling in the diffraction image. Each time the grid size is doubled, the pupil sampling period (the distance between points in the pupil) decreases by the square root of 2 in each dimension, the image surface sampling period also decreases by the square root of 2 in each dimension, and the width of

the diffraction image grid increases by a factor of the square root of 2 (since there are twice as many points in each dimension). All ratios are approximate, and asymptotically correct for large grids.

The stretching is referenced to a grid size of 32 x 32. The 32 x 32 grid of points is placed over the pupil, and the points that lie within the pupil are actually traced. For this grid size, the default distance between points in the diffraction image surface is given by

$$\Delta = \lambda F \left(\frac{n-2}{2n} \right)$$

where F is the working F/# (not the same as the image space F/#), $\square$ is the shortest defined wavelength, and n is the number of points across the grid. The -2 factor is due to the fact the pupil is not centered on the grid (since n is even), but is offset at n/2 + 1. The 2n in the denominator is due to the zero-padding described later.

For grids larger than 32 x 32, the grid is by default stretched in pupil space by a factor of $\sqrt{2}$ each time the sampling density doubles. The general formula for the sampling in image space is then

$$\Delta = \lambda F \left(\frac{n-2}{2n} \right) \left[\frac{32}{n} \right]^{1/2}$$

and the total width of the image data grid is

$$W = 2n\Delta$$

Since the stretching of the pupil grid decreases the number of sample points in the pupil, the effective grid size (the size of the grid that actually represents traced rays) is smaller than the sampling grid. The effective grid size increases as the sampling increases, but not as quickly. The following table summarizes the approximate effective grid size for various sampling density values.

DEFAULT EFFECTIVE GRID SIZES FOR PSF CALCULATIONS

Sampling Grid Size	Approximate Effective Pupil Sampling
32 x 32	32 x 32
64 x 64	45 x 45
128 x 128	64 x 64
256 x 256	90 x 90
512 x 512	128 x 128
1024 x 1024	181 x 181
2048 x 2048	256 x 256
4096 x 4096	362 x 362
8192 x 8192	512 x 512

The sampling is also a function of wavelength. The discussion above is only valid for the shortest wavelength used in the calculation. If the computation is polychromatic, then the longer wavelengths will be scaled to have smaller effective grids. The scale factor used is the ratio of the wavelengths. This should be considered when selecting sampling grids for systems with broad wavelength bands. For polychromatic computations, the data for shorter wavelengths is more accurate than for longer wavelengths. For the FFT PSF phase, real, and imaginary data types, the polychromatic average is not physically meaningful, and Zemax will set the wavelength selection to the primary wavelength if the user selected all wavelengths.

The image delta, Δ , can be selected manually if a different sampling distance is required. If the image delta is zero, Zemax uses the default spacing and sampling grids described above. If the image delta is greater than zero, then Zemax scales the pupil sampling to yield the desired image delta size. The actual amount of stretching depends upon the grid size, the image delta, the defined wavelengths, the F/#'s at each field and wavelength, and the aspect ratio of the exit pupil. If the image delta is set too small, then not enough points will be left to sample the pupil; if the image delta is too big, then the pupil grid will not extend over the full width of the exit pupil. Both of these cases are trapped by Zemax and an error message will be issued if they occur. If the image delta is less than zero, then Zemax does not stretch the pupil at all. This maximizes the extent of the PSF in image coordinates and uses the full grid of rays in the pupil, at the expense of not increasing the spatial resolution with sampling, as the image delta will be nearly constant with increased sampling.

Once the sampling is specified, Zemax doubles the array size in a process called "zero padding". This means for a 32 x 32 sampling, Zemax uses the center portion of a 64 x 64 grid. Therefore, the diffraction PSF will be distributed over a 64 x 64 size grid. The sampling in the image space is always twice the pupil sampling. Zero padding is performed to reduce aliasing.

The nature of the FFT algorithm is that the computation is done in pupil space coordinates. For this reason, rotating the image surface will have no effect on the orientation of the computed PSF. The X and Y orientation of the PSF corresponds to the X and Y orientation of rays in the entrance pupil, which is not always the same as the spatial X and Y orientation of the image. To compute the PSF in the image space coordinates, see "Huygens PSF"

FFT Cross Section

This feature plots cross sections through the diffraction PSF.
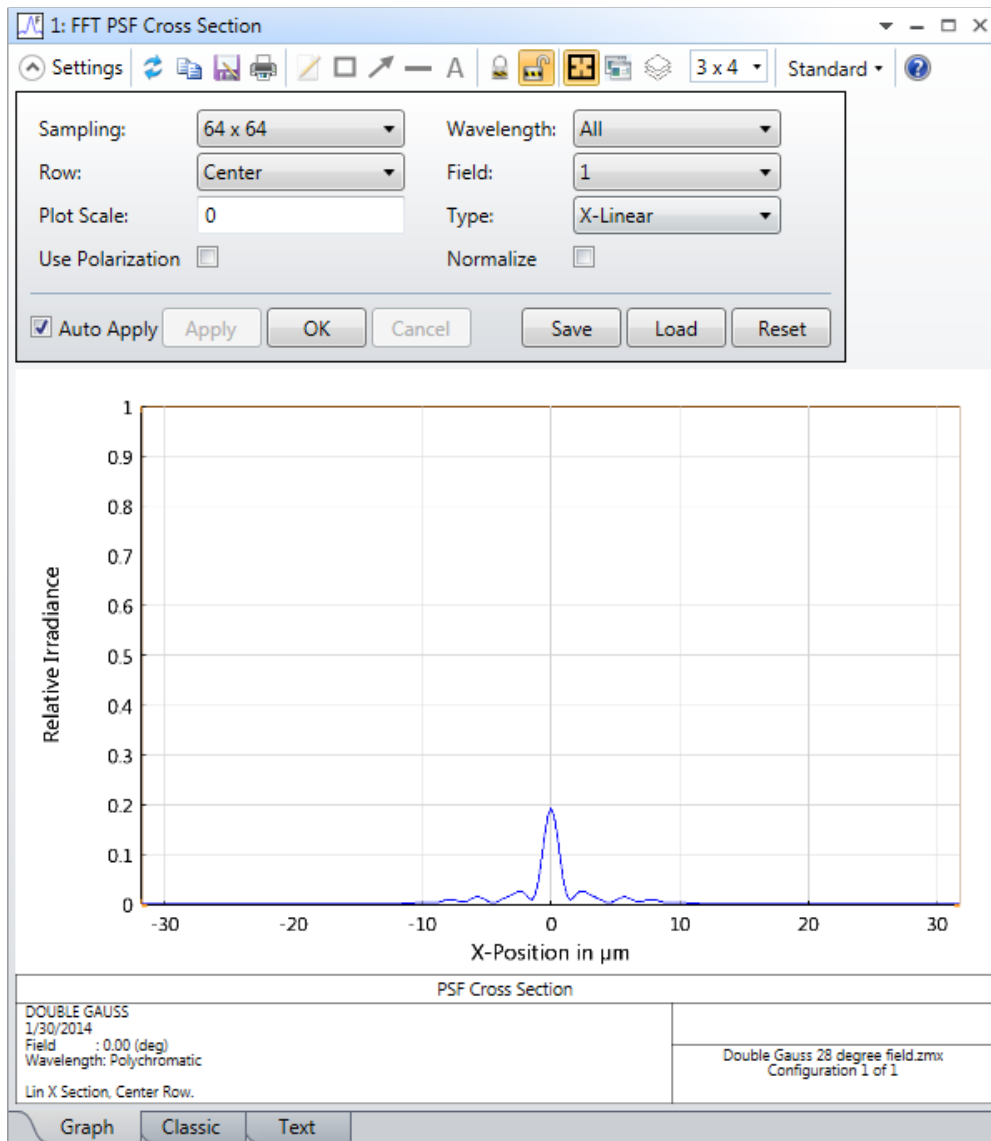
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Row/Col The row or column to display. For a sampling of 32 x 32, there are 64 rows and 64 columns (see the discussion section of the FFT PSF feature). Whether a row or a column is used depends upon the "Type" setting. The default "Center" displays the center of the PSF, which is not always the peak of the PSF.

Plot Scale Sets the maximum horizontal scale in micrometers. Enter zero for automatic scaling.

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Type Select X or Y cross sections, either linear, logarithmic, phase, real part, or imaginary part. X cross sections are called rows, and Y cross sections are called columns.

Use Polarization If checked, polarization is considered. See "Polarization for information on defining the polarization state and how polarization is used by analysis features.

Normalize If checked, the peak intensity will be normalized to unity. Otherwise, the peak intensity is normalized to the peak of the unaberrated PSF (the Strehl ratio).

Discussion The cross sections are taken directly from the PSF data. Because the PSF is computed directly from the phase in the exit pupil, the orientation of the coordinate system may not be correct in all cases. What constitutes positive x or y may not agree with data presented in image space coordinates such as the spot diagram.

Note: Also see the discussion section of the FFT PSF. Those comments also apply to this feature.

FFT Line/Edge Spread

This feature plots the edge or line spread functions based upon a computation and integration of the diffraction FFT PSF.
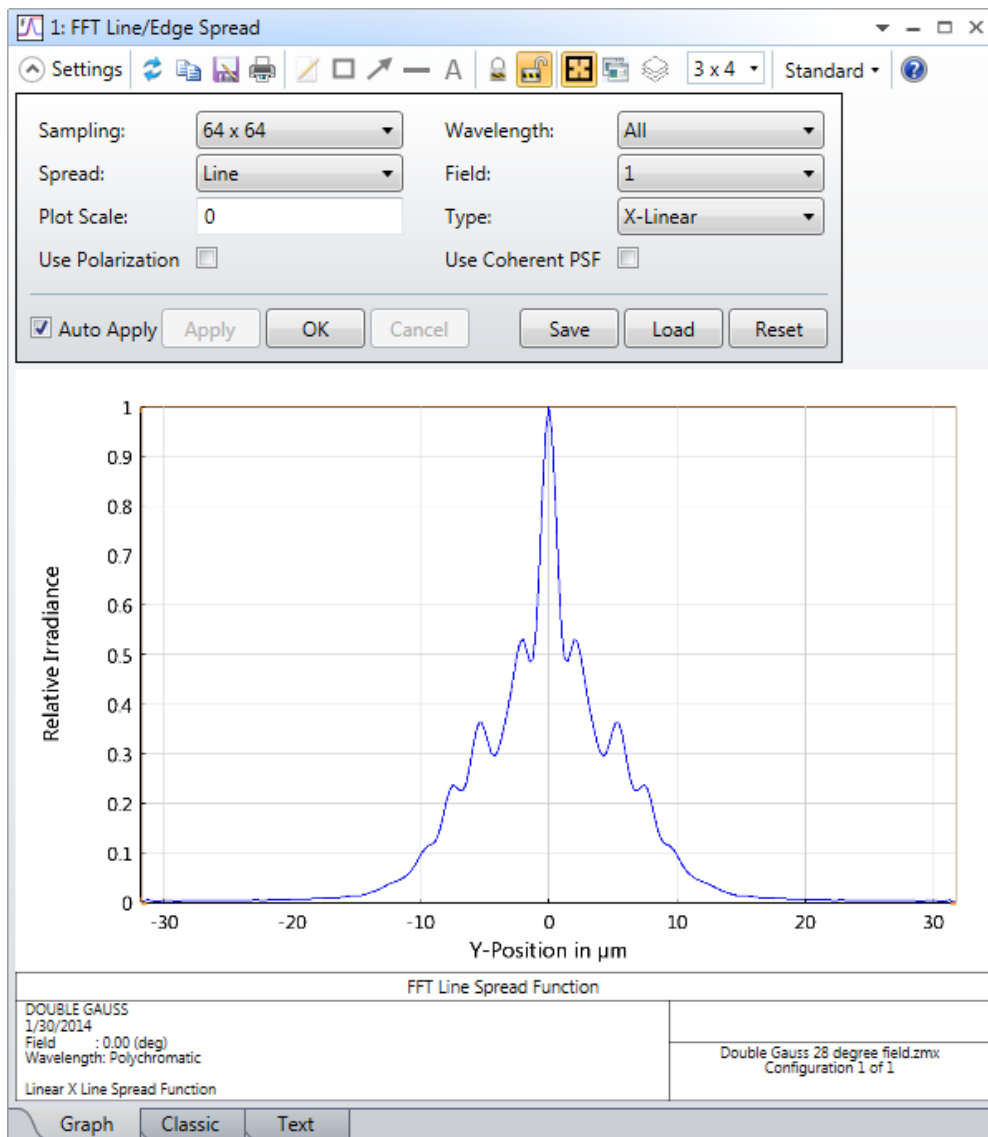
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Spread Choose either line or edge spread function.

Plot Scale Sets the maximum horizontal scale in micrometers. Enter zero for automatic scaling.

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Type Select X or Y orientation, and either linear intensity or logarithmic intensity.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Coherent PSF If checked, the computation is done assuming coherent illumination. If unchecked, the illumination is considered incoherent.

Discussion

The line response function (or line spread function, LSF) is the cross section of the image of a line object. The edge spread function (ESF) is the cross section of the image of an edge (a semi-infinite plane). The X- or Y- orientation refers to the direction of the line or edge. The X-orientation means the line or edge is parallel to the X-axis. The Y-orientation means the line or edge is parallel to the Y-axis. The LSF and ESF are computed by integration of the FFT PSF. See also "Geometric Line/Edge Spread"

Note: Also see the discussion section of the FFT PSF. Those comments also apply to this feature.

Huygens PSF

Computes the diffraction PSF using direct integration of Huygens wavelets method. The Strehl ratio is also computed.
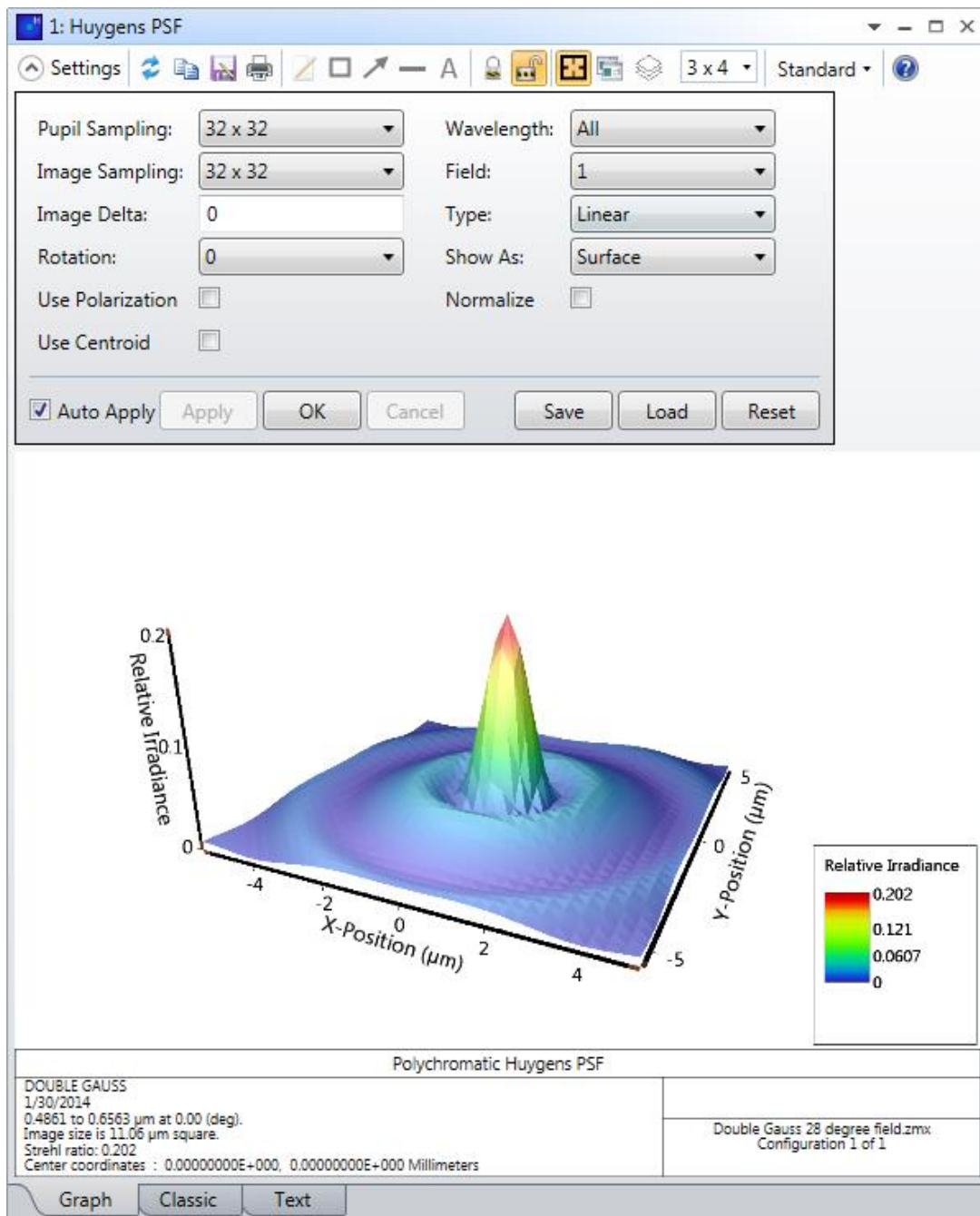
Pupil Sampling Selects the size of the grid of rays to trace to perform the computation. Higher sampling densities yield more accurate results at the expense of longer computation times.

Image Sampling The size of the grid of points on which to compute the diffraction image intensity. This number, combined with the image delta, determine the size of the area displayed.

Image Delta The distance in micrometers between points in the image grid. Use zero for the default grid spacing.

Rotation Rotation specifies how the surface plots are rotated; either 0, 90, 180, or 270 degrees.

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Type Select linear (intensity), or logarithmic (intensity). The logarithmic scaling can range from

1 to 5 decades. Also available are real amplitude, imaginary amplitude, and phase in degrees.

Configuration Select "All" to perform a coherent sum of the PSF at each wavelength across all configurations, or select the "Current" or any single configuration. Note this is a coherent sum for the same wavelength in each configuration, followed by an incoherent sum of the resulting PSF's for different wavelengths. For this reason, each defined wavelength must be the same in all configurations. Wavelength and configuration weights may be used but the wavelength values must be identical. See CWGT and WLWT in "Summary of multi-configuration operands". This coherent sum also assumes that the image surface is located in the identical position in all configurations. If "All" is selected, and both focal and afocal mode configurations are defined, the Configuration setting will automatically be reset to "Current".

Normalize If checked, the peak intensity will be normalized to unity. Otherwise, the peak intensity is normalized to the peak of the unaberrated PSF (the Strehl ratio).

Show As Choose surface plot, contour map, grey scale, or false color map as the display option. The True Color option creates an RGB color representation of the PSF by converting the wavelengths to the closest RGB equivalent and summing over all wavelengths. The accuracy of the True Color presentation is limited by the RGB method of rendering color on a computer display; and it is not possible to exactly represent monochromatic colors. The True Color option cannot be used if the Type is anything other than linear. Note that choosing True Color will require a recalculation of the data, as it is computed by a separate method. Choosing between the other display options does not require a recalculation.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.
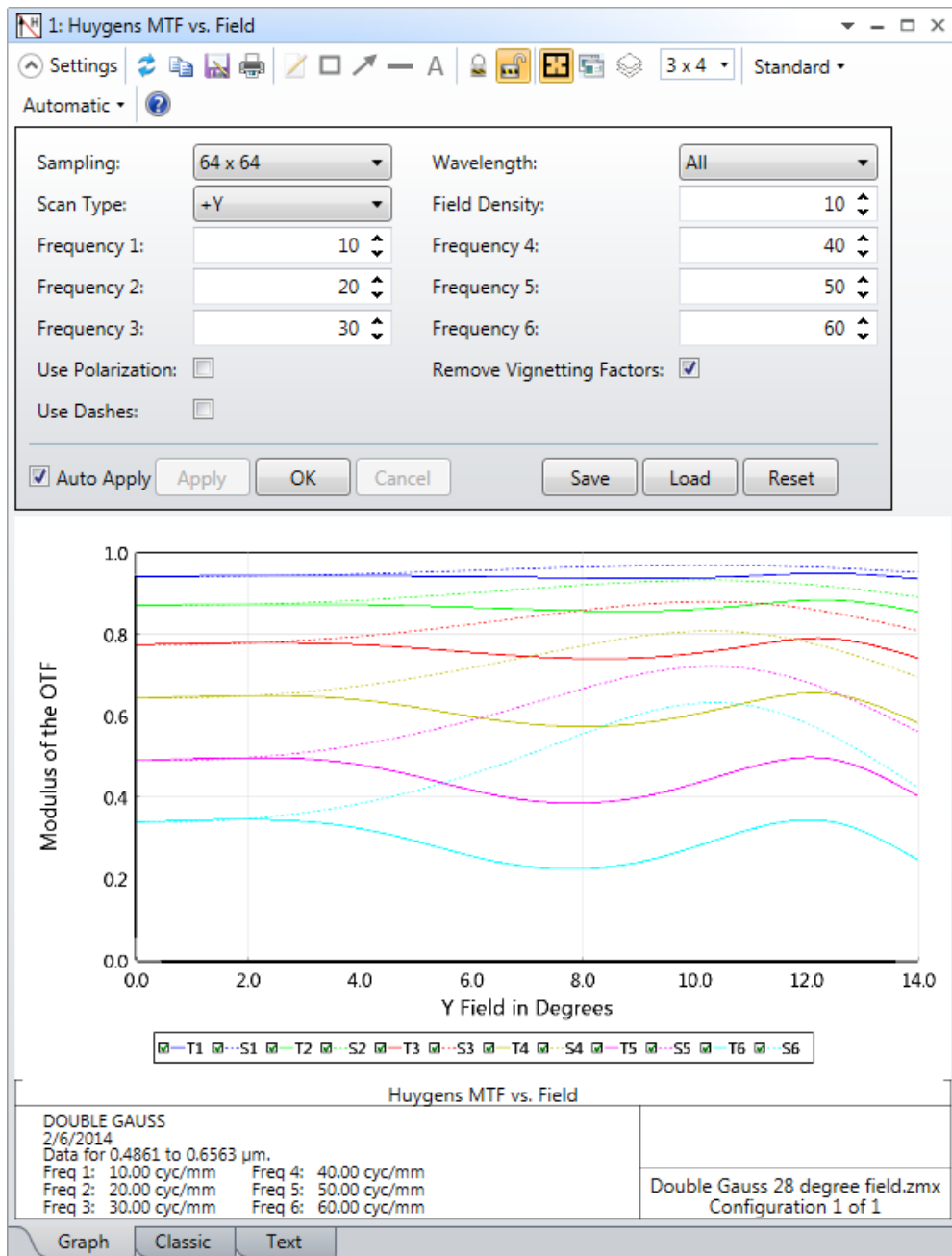
Use Centroid If checked, the plot will be centered on the geometric image centroid. If unchecked, the plot will be centered on the chief ray.

Discussion

One way of considering the effects of diffraction is to imagine each point on a wavefront as a perfect point source with an amplitude and phase. Each of these point sources radiates a spherical "wavelet", sometimes called a "Huygens wavelet" after Huygens, who first proposed the model. The diffraction of the wavefront as it propagates through space is given by the interference, or complex sum, of all the spherical wavelets radiated.

To compute the Huygens PSF, a grid of rays is launched through the optical system, and each ray represents a particular amplitude and phase wavelet. The diffraction intensity at any point on the image surface is the complex sum of all these wavelets, squared. The PSF is computed this way for every point on the image grid. The center point listed in the text output of the PSF calculation is defined as the coordinate $(n/2+1)$, $(n/2+1)$, where n is the number of points in the image space grid. The data output for a rotationally symmetric system will be symmetric about the point $(n/2, n/2+1)$.

The Image Delta value determines the point spacing of the image space grid. If a value of zero is specified, a default grid spacing is used. The default Image Delta is given by:

$$\Delta = \frac{\lambda F}{\sqrt{n_p}}$$

where n_p is the number of points in the pupil space grid, λ is the longest wavelength used in the calculation, and F is the working F/#. The exact value of the image Delta size is not critical, as long as the entire width of the PSF is included within the range of n * (Image Delta).

Unlike the FFT PSF, Zemax computes the Huygens PSF on an imaginary plane tangent to the image surface at the chief ray intercept. Note the imaginary plane is normal to the normal of the surface, not the chief ray. Therefore, the Huygens PSF accounts for any local tilt in the image surface caused by either the image surface slope, the chief ray incidence angle, or both.

The Huygens method accounts for the evolving shape of the diffraction image as the beam propagates along the image surface. This is an important effect if the image surface is tilted with respect to the incoming beam. Another advantage to the Huygens PSF method is that any grid size and spacing may be selected by the user. This allows direct comparison between PSF's from two different lenses, even if the F/#'s or wavelengths are different.

The only disadvantage of the Huygens PSF is speed. Direct integration is slow when compared to the FFT method (see the previous section for details). The computation time depends upon the pupil grid size squared times the image grid size squared, times the number of wavelengths. Zemax accounts for any symmetry the system has. The Huygens PSF automatically uses all available processors for maximum speed on multiple CPU computers.

Huygens Cross Section

Computes the diffraction PSF using direct integration of Huygens wavelets method. The Strehl ratio is also computed. This feature is very similar to the "Huygens PSF", with the difference being the data is plotted as a cross section.
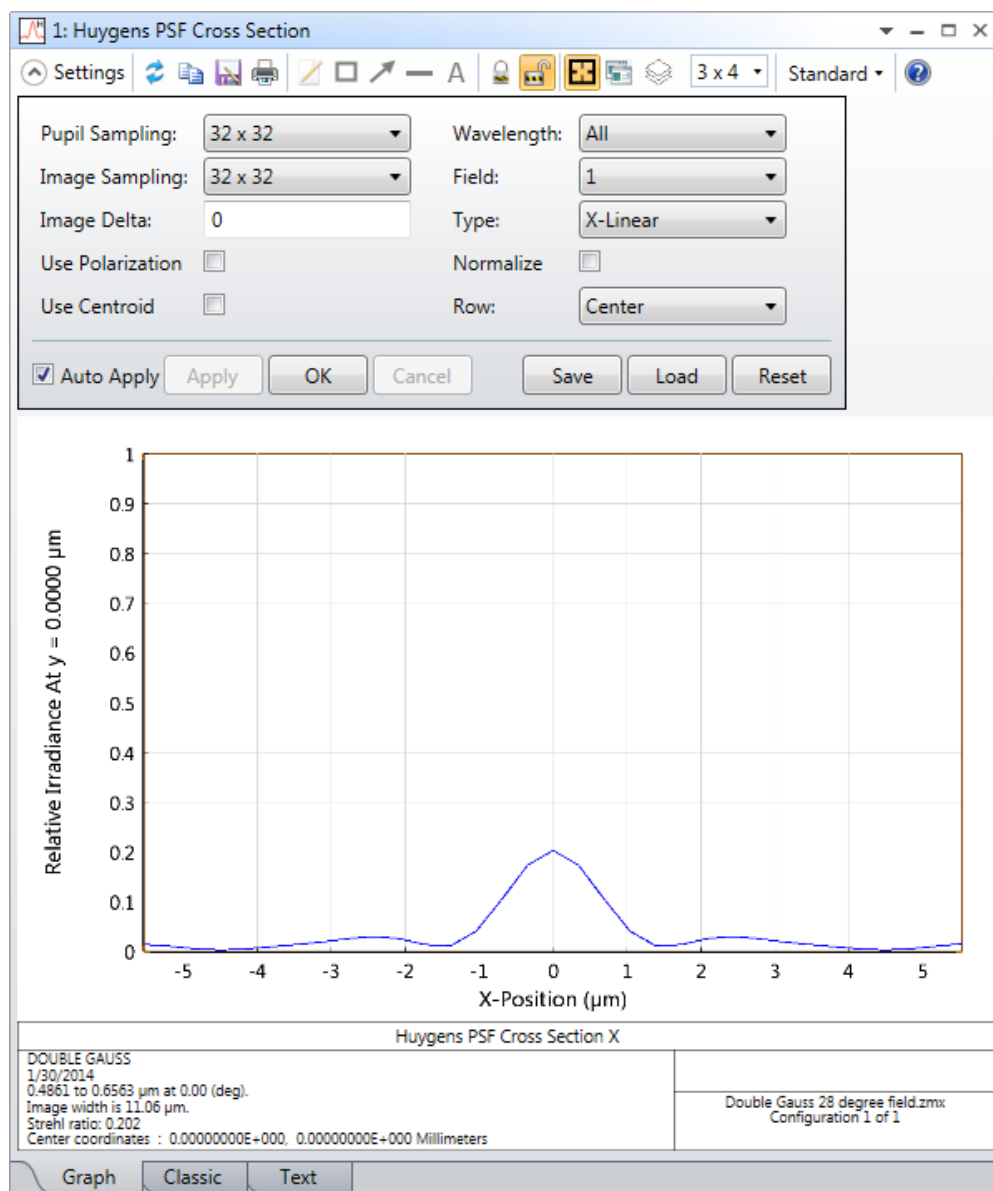
MTF
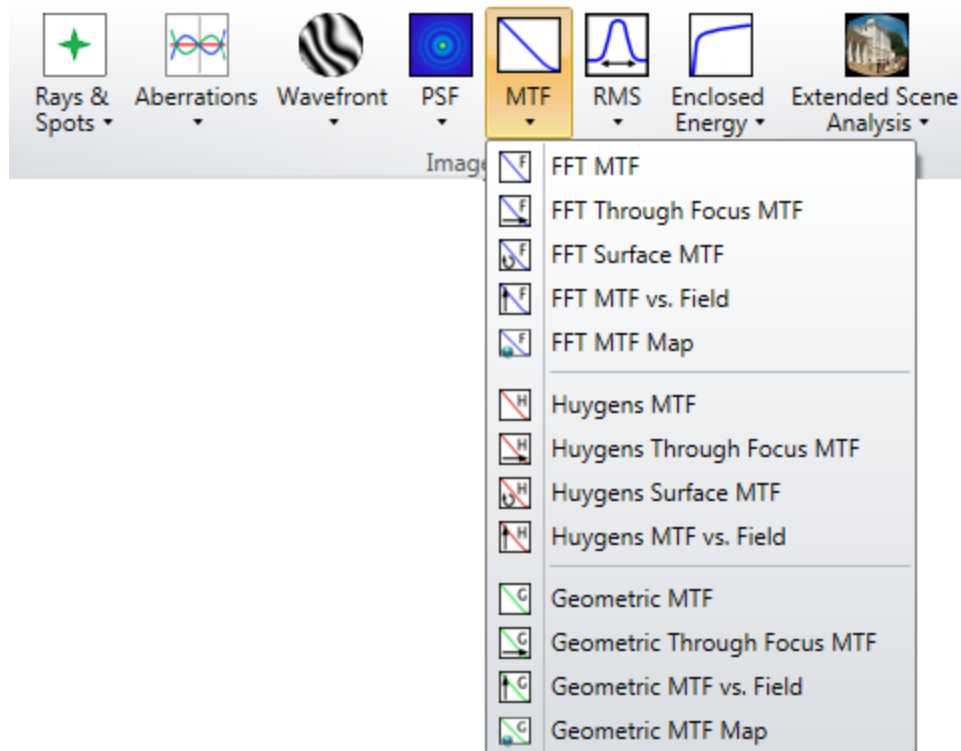
Various calculations of the Modulation Transfer Function (MTF) of a lens system

FFT MTF

Computes the diffraction modulation transfer function (MTF) data for all field positions using an FFT algorithm.
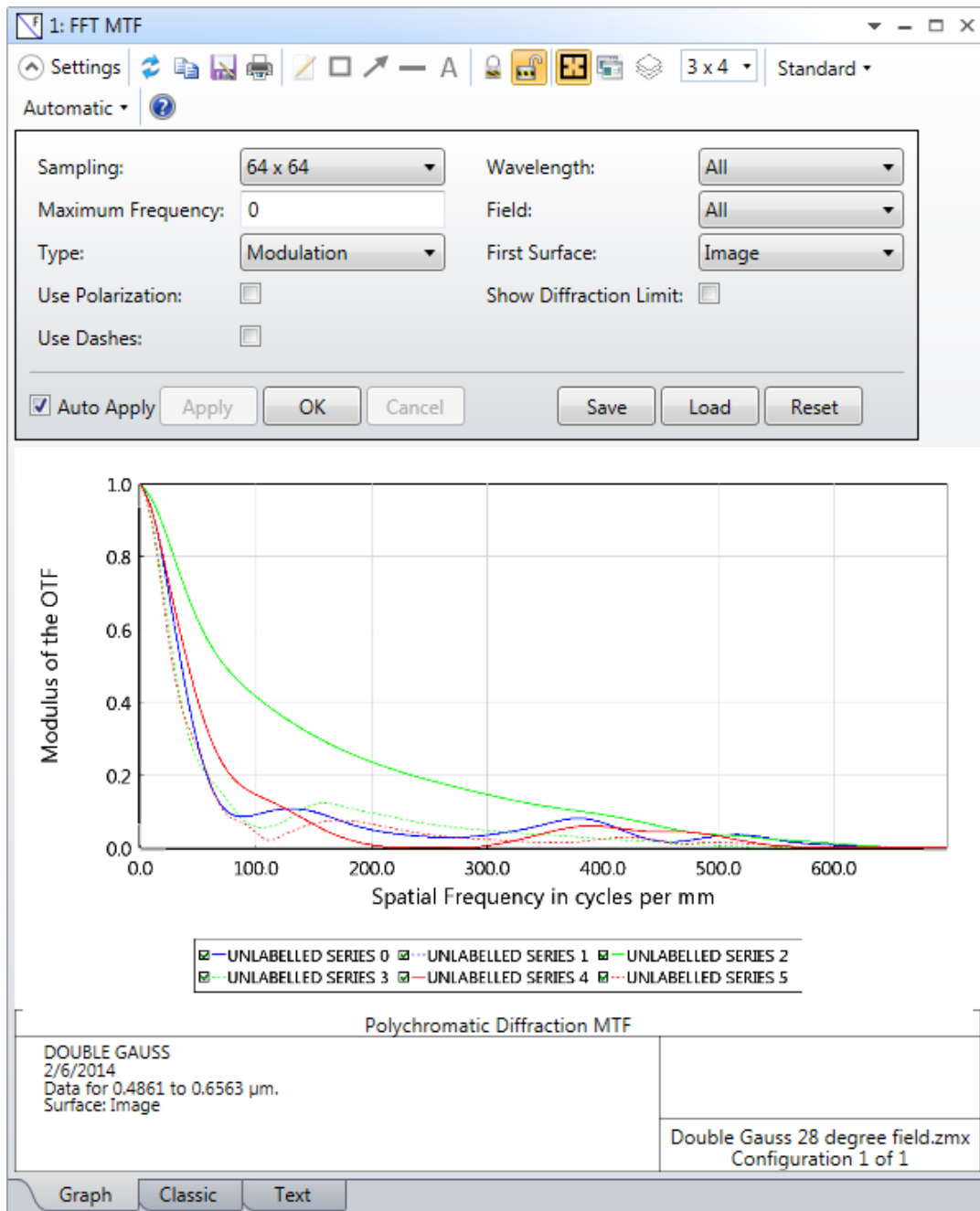
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Show Diffraction Limit Select whether or not the diffraction limited data should be displayed.

Max Frequency Specify the maximum frequency (see "MTF Units") plotted.

Wavelength The wavelength number to be used in the calculation.

Field The field number to be used in the calculation.

Type Select either modulation, real, imaginary, phase, or square wave response.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Surface Selects the surface at which the MTF is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Discussion

Note: See the discussion sections of the FFT and Huygens PSF. Those comments also apply to this feature.

The diffraction MTF computation is based upon an FFT of the pupil data. The resulting MTF is the modulation as a function of spatial frequency for a sine wave object, although optionally the real, imaginary, phase, or square wave response is available. The Square wave MTF is the modulation response for a square wave target of the specific frequency, as opposed to the response to a sine wave target for the other plots. The square wave response is computed from the MTF data using the following formula:

$$S(v) = \frac{4}{\pi}\left[\frac{M(v)}{1} - \frac{M(3v)}{3} + \frac{M(5v)}{5} - \frac{M(7v)}{7}...\right]$$

where S(v) is the square wave response, M(v) is the sinusoidal modulation response, and v is the spatial frequency.

For focal systems, the cutoff frequency at any one wavelength is given by one over the wavelength times the working F/#. Zemax computes the working F/# at each wavelength for each field for the sagittal and tangential response separately. This yields accurate MTF data even for systems with anamorphic and chromatic distortion, such as those incorporating cylinders or gratings. For afocal systems, the cutoff frequency is the exit pupil diameter divided by the wavelength.

The diffraction calculations are more accurate as the sampling increases, the peak-to-valley and maximum slope of the OPD decrease, and the transverse ray aberrations decrease. If the peak-to-valley OPD in the pupil is too great, then the wavefront sampling is too coarse and aliasing occurs. Aliasing will result in inaccurate data. Zemax will attempt to detect when aliasing occurs, and issue an appropriate error message. However, Zemax cannot automatically detect when the sampling is too low in all cases, especially in the presence of very steep slopes on the wavefront phase.

The FFT based MTF assumes a (reasonably) uniform distribution of rays on the exit pupil in cosine space to be accurate. Some systems, such as extremely fast off-axis reflectors, have dramatic stretching of the exit pupil and the FFT based MTF will thus be inaccurate. For these systems, the Huygens MTF should be used instead. For more information, see "Huygens MTF".

When the OPD in waves is large, such as more than 10 waves, it is probably a good idea to switch to geometric MTF instead of diffraction MTF. For these highly aberrated systems, the geometric MTF is very accurate, especially at low spatial frequencies (the higher frequency MTF falls of rapidly when aberrations are large). If shown, the diffraction limit curve is for the aberration free response at the reference field position; see "Diffraction limited".

The spatial frequency scale of the MTF plot is in cycles per mm in image space, or cycles per milliradian in object space (see "MTF Units"). MTF in cycles per mm is computed in image space, so any magnification of the system needs to be considered when determining spatial frequency response for object space.

The nature of the FFT algorithm is that the computation is done in pupil space coordinates. For this reason, rotating the image surface will have no effect on the orientation of the computed MTF. The tangential response corresponds to a periodic target oriented with lines along the object space X axis, and the sagittal response corresponds to a periodic target oriented with lines along the object space Y axis. This is different from the conventions of the "Geometric MTF" and the "Huygens MTF".

FFT Through Focus MTF

Computes the FFT modulation transfer function data as a function of focus shift at a specific frequency.

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Delta Focus For focal systems, delta focus is the ± Z-axis range of the plot in lens units. For afocal systems, delta focus is the ± optical power range of the plot in diopters.

Frequency The spatial frequency (see "MTF Units") for which data is plotted.

 # Steps The number of focal planes at which the data is computed. A smooth curve is drawn through the computed points. More steps yield higher accuracy and longer computation times.

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Type Select either modulation, real, imaginary, phase, or square wave response.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Discussion See "FFT MTF" for details

FFT Surface MTF

Displays the FFT computed MTF data as a 3D surface, contour, grey scale or false color map. This plot is useful for visualizing the MTF response for object orientations other than purely sagittal or tangential.
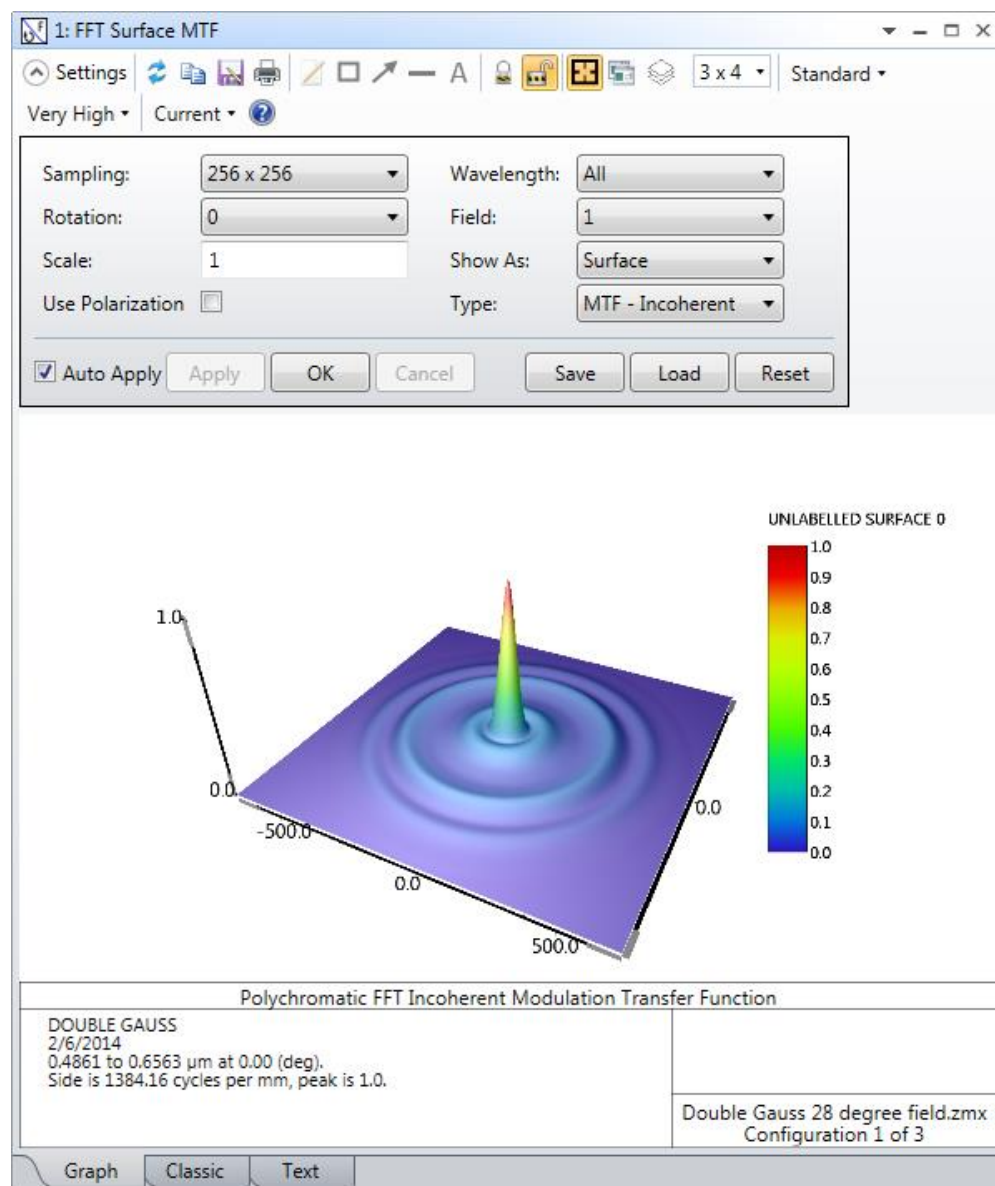
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Rotation Rotation specifies how the surface plots are rotated for viewing; either 0, 90, 180, or 270 degrees.

Scale The scale factor overrides the automatic vertical scaling set by the program on the surface plots. Generally, this value should be set to unity. The scale factor can be greater than unity to vertically stretch the plot, or less than unity to compress it.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Wavelength The wavelength number to be used in the calculation.

Field The field number determines for which defined field position the calculation should be performed.

Show As Choose surface plot, contour map, grey scale, or false color map as the display option.

Type Choose modulation, real part, or imaginary part of either the incoherent or coherent optical transfer function. If a text listing is generated and either the real or the imaginary part is requested, both the real and the imaginary parts are listed for completeness.

Discussion The regular MTF plot is just two orthogonal cross sections through the surface MTF plot. This plot is primarily qualitative. See "FFT MTF" for details.

FFT MTF vs. Field

Computes the FFT MTF data as a function of field position, and displays the data in a graph.

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Frequency 1-6 The spatial frequencies (see "MTF Units") for which data is plotted.

Wavelength The wavelength number to be used in the calculation.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See the comments about vignetting factors in the discussion section below.

Field Density The field density is the number of points between zero degrees and the maximum field at which the MTF is calculated, intermediate values are interpolated. A maximum of 100 field points is allowed.

Scan Type Choose +y, +x, -y, or -x field scan direction.

Discussion See "FFT MTF" for details. This feature plots MTF vs. field height up to the maximum defined radial field coordinate.

Comment about vignetting factors

Vignetting factors determine the size and shape of the pupil as seen from different field points (see "Vignetting factors" for a full discussion). Because this analysis feature needs to trace rays at arbitrary intermediate field points where no specific vignetting factors are defined, the use of vignetting factors is not recommended. If "Remove Vignetting Factors" is checked on (the default), any defined vignetting factors will automatically be replaced with surface apertures for this computation. The surface aperture method is generally more accurate than the vignetting factor method when the pupil is overfilled with light. The resulting data may be different between the two methods. In some cases, particularly where the vignetting factors are being used to define the shape of the source beam rather than the apertures of the optics, it may be required to use the defined vignetting factors. In this case, check the "Remove Vignetting Factors" box off. Zemax will then use the closest defined field to determine the vignetting factors to use for an arbitrary field point.

FFT MTF Map

Computes the FFT MTF as a function of field position, and displays the data over a rectangular region of field.

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

X or Y Field Width The X or Y field width in field units. This is the total width or height, not the half width or height. Field units are degrees in object space if field angle is used, otherwise field units are the same as lens units.

Frequency The spatial frequency (see "MTF Units") for which data is plotted.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Wavelength The wavelength number to be used in the calculation, or All for polychromatic MTF.

X or Y Pixels The number of pixels at which to compute the MTF in each respective direction. Note the size of the pixels is determined by both the number of pixels and the width of the field; the pixels are not required to be square. The MTF is computed at the center of the pixel and the MTF is assumed to have that value over the entire region of the pixel for display purposes.

MTF Data Choose Tangential, Sagittal, or average MTF to be displayed.

Reference Field This control selects the field number that corresponds to the center of the map. If zero is selected the (0, 0) field coordinate is used as the center of the map.

Show As Choose grey scale or false color map as the display option.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Discussion See "FFT MTF" for details. This feature computes the MTF at each field point on a 2D grid. If the total number of points is large, the computation time may become quite large. See also the "Geometric MTF Map"

Comment about vignetting factors

Vignetting factors determine the size and shape of the pupil as seen from different field points (see "Vignetting factors" for a full discussion). Because this analysis feature needs to trace rays at arbitrary intermediate field points where no specific vignetting factors are defined, the use of vignetting factors is not recommended. If "Remove Vignetting Factors" is checked on (the default), any defined vignetting factors will automatically be replaced with surface apertures for this computation. The surface aperture method is generally more accurate than the vignetting factor method when the pupil is overfilled with light. The resulting data may be different between the two methods. In some cases, particularly where the vignetting factors are being used to define the shape of the source beam rather than the apertures of the optics, it may be required to use the defined vignetting factors. In this case, check the "Remove Vignetting Factors" box off. Zemax will then use the closest defined field to determine the vignetting factors to use for an arbitrary field point.

Huygens MTF

Computes the diffraction modulation transfer function (MTF) data using a Huygens direct integration algorithm.

Pupil Sampling Selects the size of the grid of rays to trace to perform the computation. Higher sampling densities yield more accurate results at the expense of longer computation times.

Image Sampling The size of the grid of points on which to compute the diffraction image intensity. This number, combined with the image delta, determine the size of the PSF used to compute the MTF. See also "Huygens PSF" .

Image Delta The distance in micrometers between points in the image grid. Use zero for the default grid spacing.

Configuration Select "All" to perform a calculation of the MTF based upon a coherent sum of the PSF at each wavelength across all configurations, or select the "Current" or any single configuration. Note the MTF is computed from the PSF, which is a coherent sum for the same wavelength in each configuration, followed by an incoherent sum of the resulting PSF's for different wavelengths. For this reason, each defined wavelength must be the same in all configurations. Wavelength and configuration weights may be used but the wavelength values must be identical. See CWGT and WLWT in "Summary of multi- configuration operands. This coherent sum also assumes that the image surface is located in the identical position in all configurations. If "All" is selected, and both focal and afocal mode configurations are defined, the Configuration setting will automatically be reset to "Current".

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Type Select the data to display, currently modulation is the only option.

Max Frequency The maximum frequency in MTF units (see "MTF Units") to display.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Discussion

See the discussion sections of the Huygens PSF. Those comments also apply to this feature.

The Huygens MTF computes an FFT of the Huygens PSF. The settings for Image Sampling and Image Delta are the same as for the Huygens PSF, therefore it is instructive to do a Huygens PSF first (see "Huygens PSF"). Since the transform is done on the PSF in image space coordinates, the tangential response corresponds to spatial frequencies in the y direction in local image surface coordinates, and the sagittal response corresponds to spatial frequencies in the x direction. The Huygens MTF also has no dependence on the location of rays in the paraxial pupils. The MTF can therefore be computed for any system that the Huygens PSF can be computed for including many non-sequential systems using ports where reference rays required by other diffraction algorithms would not make it through, or for systems where pupils or images formed by multiple non- sequential sub-apertures are overlapped. Systems with extreme exit pupil distortion, such as very fast off-axis reflectors, are also handled correctly with the Huygens technique.

The nature of the Huygens algorithm is that the computation is done in image space coordinates. For this reason, rotating the image surface will affect the orientation of the computed MTF. The tangential response corresponds to the image of a periodic target oriented with lines along the image space X axis, and the sagittal response corresponds to the image of a periodic target oriented with lines along the image space Y axis. This is different from the conventions of the "FFT MTF"

Huygens Through Focus MTF

Computes the diffraction modulation transfer function (MTF) data using a Huygens direct integration algorithm and displays the data as a function of delta focus.

Pupil Sampling Selects the size of the grid of rays to trace to perform the computation. Higher sampling densities yield more accurate results at the expense of longer computation times.

Image Sampling The size of the grid of points on which to compute the diffraction image intensity. This number, combined with the image delta, determine the size of the PSF used to compute the MTF. See also "Huygens PSF".

Image Delta The distance in micrometers between points in the image grid. Use zero for the default grid spacing.

Configuration Select "All" to perform a calculation of the MTF based upon a coherent sum of the PSF at each wavelength across all configurations, or select the "Current" or any single configuration. Note the MTF is computed from the PSF, which is a coherent sum for the same wavelength in each configuration, followed by an incoherent sum of the resulting PSF's for different wavelengths. For this reason, each defined wavelength must be the same in all configurations. Wavelength and configuration weights may be used but the wavelength values must be identical. See CWGT and WLWT in "Summary of multi- configuration operands". This coherent sum also assumes that the image surface is located in the identical position in all configurations. If "All" is selected, and both focal and afocal mode configurations are defined, the Configuration setting will automatically be reset to "Current".

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Type Select the data to display, currently modulation is the only option.

Max Frequency The maximum frequency in MTF units (see "MTF Units") to display.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Delta Focus For focal systems, delta focus is the ± Z-axis range of the plot in lens units. For afocal systems, delta focus is the ± optical power range of the plot in diopters.

Frequency The spatial frequency (see "MTF Units") for which data is plotted.

 # Steps The number of focal planes at which the data is computed. A smooth curve is drawn through the computed points. More steps yield higher accuracy and longer computation times.

Discussion This feature is very similar to the Huygens MTF feature (see "Huygens MTF").

Huygens Surface MTF

Computes the diffraction modulation transfer function (MTF) data using a Huygens direct integration algorithm and displays the data as a grey scale or false color plot.

Pupil Sampling Selects the size of the grid of rays to trace to perform the computation. Higher sampling densities yield more accurate results at the expense of longer computation times.

Image Sampling The size of the grid of points on which to compute the diffraction image intensity. This number, combined with the image delta, determine the size of the PSF used to compute the MTF. See also "Huygens PSF" .

Image Delta The distance in micrometers between points in the image grid. Use zero for the default grid spacing.

Configuration Select "All" to perform a calculation of the MTF based upon a coherent sum of the PSF at each wavelength across all configurations, or select the "Current" or any single configuration. Note the MTF is computed from the PSF, which is a coherent sum for the same wavelength in each configuration, followed by an incoherent sum of the resulting PSF's for different wavelengths. For this reason, each defined wavelength must be the same in all configurations. Wavelength and configuration weights may be used but the wavelength values must be identical. See CWGT and WLWT in "Summary of multi- configuration operands" . This coherent sum also assumes that the image surface is located in the identical position in all configurations. If "All" is selected, and both focal and afocal mode configurations are defined, the Configuration setting will automatically be reset to "Current".

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Type Select the data to display, currently modulation is the only option.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Show As Choose grey scale or false color map as the display option.

Type Choose modulation, real part, or imaginary part of either the incoherent or coherent optical transfer function. If a text listing is generated and either the real or the imaginary part is requested, both the real and the imaginary parts are listed for completeness.

Discussion This feature is very similar to the Huygens MTF feature (see "Huygens MTF").

Huygens MTF vs. Field

Computes the Huygens MTF data as a function of field position, and displays the data in a graph.

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Frequency 1-6 The spatial frequencies (see "MTF Units") for which data is plotted.

Wavelength The wavelength number to be used in the calculation.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See the comments about vignetting factors in the discussion section below.

Field Density The field density is the number of points between zero degrees and the maximum field at which the MTF is calculated, intermediate values are interpolated. A maximum of 100 field points is allowed.

Scan Type Choose +y, +x, -y, or -x field scan direction.

Discussion This feature is very similar to the "FFT MTF vs. Field", except the Huygens method is used to compute the MTF rather than the FFT method. See "Huygens MTF" for details on the Huygens MTF method.

Geometric MTF

Computes the geometric MTF, which is an approximation to the diffraction MTF based upon ray aberration data.
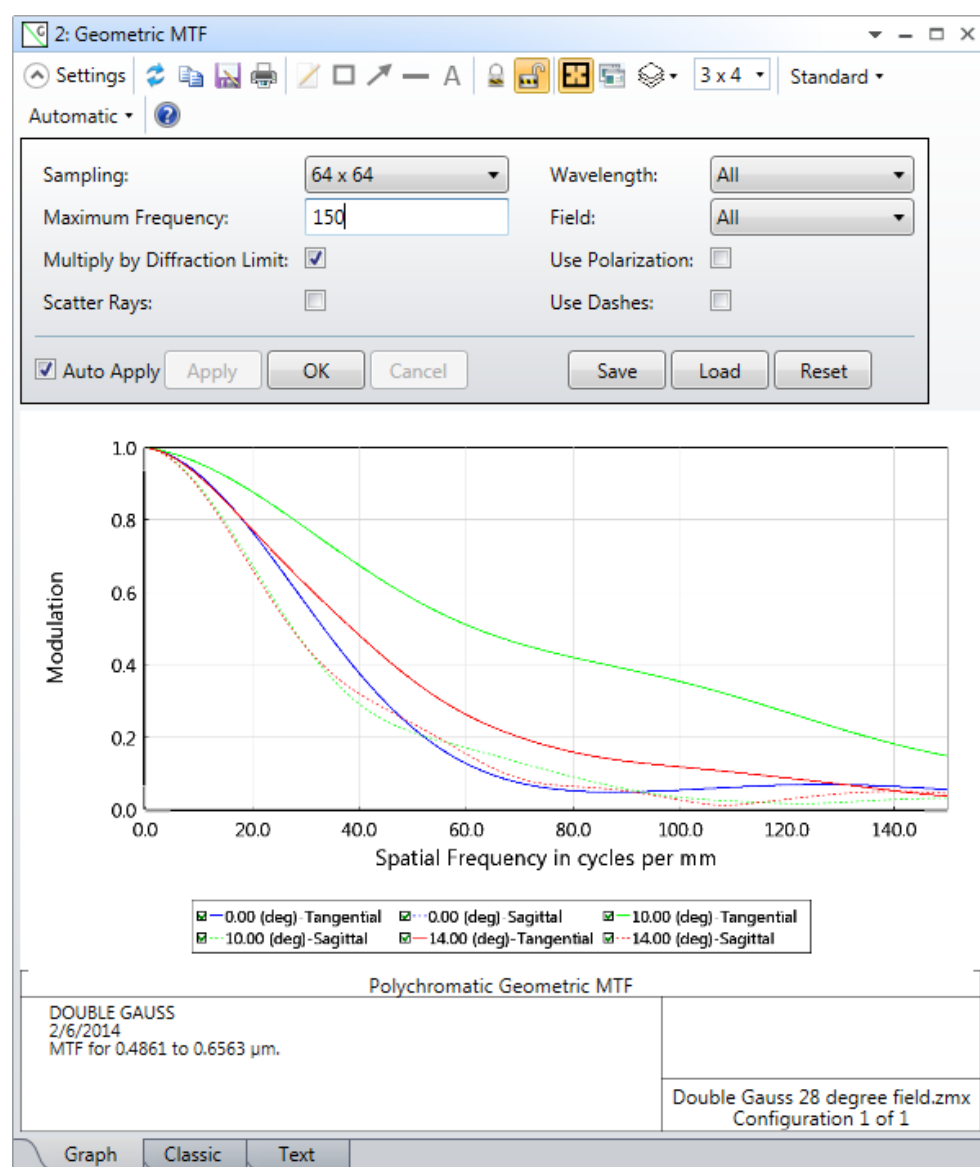
Sampling   The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Max Frequency The maximum spatial frequency (see "MTF Units") for which data is plotted.

Wavelength   The wavelength number to be used in the calculation.

Field   The field number for which the calculation should be performed.

Multiply by Diffraction Limit   When checked, will scale the geometric MTF by the diffraction limited MTF to yield a more realistic result for systems with small aberrations. Should always be used.

Use Polarization   If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Scatter Rays   If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. See "Surface scattering settings".

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Discussion

The geometric MTF is a useful approximation to the diffraction MTF if the system is not close to the diffraction limit. The primary advantage to using the geometric MTF is for systems which have too many waves of aberration to permit accurate calculation of the diffraction MTF. The geometric MTF is also very accurate at low spatial frequencies for systems with large aberrations.

The nature of the geometric algorithm is that the computation is done in image space coordinates. For this reason, rotating the image surface will affect the orientation of the computed MTF. The tangential response corresponds to the image of a periodic target oriented with lines along the image space X axis, and the sagittal response corresponds to the image of a periodic target oriented with lines along the image space Y axis. This is different from the conventions of the "FFT MTF"

Geometric Through Focus MTF

Computes the geometric MTF data through focus at a specific spatial frequency.

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Delta Focus For focal systems, delta focus is the ± Z-axis range of the plot in lens units. For afocal systems, delta focus is the ± optical power range of the plot in diopters.

Frequency The spatial frequency (see "MTF Units") for which data is plotted.

 # Steps The number of focal planes at which the data is computed. A smooth curve is drawn through the computed points. More steps yield higher accuracy and longer computation times.

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Multiply by Diffraction Limit When checked, will scale the geometric MTF by the diffraction limited MTF to yield a more realistic result for systems with small aberrations. Should always be used.

Use Polarization If checked, polarization is considered. See "[Polarization](#)" for information on defining the polarization state and how polarization is used by analysis features.

Scatter Rays   If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. See "[Surface scattering settings](#)" .

Use Dashes   Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Discussion See the "Geometric MTF" for details.

Geometric MTF vs. Field

Computes the geometric modulation transfer function data as a function of field position.
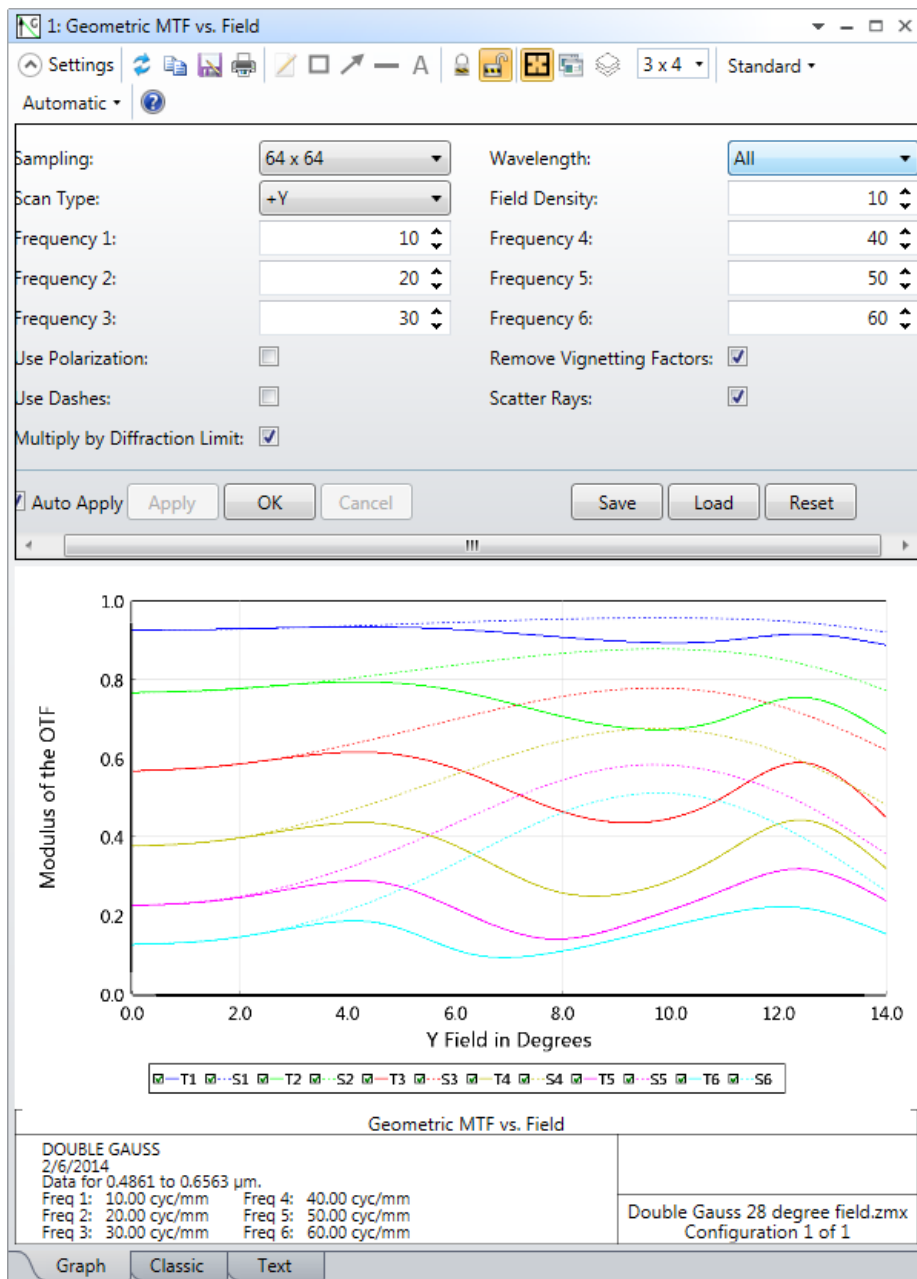
Settings The settings are identical to those for the FFT MTF vs. Field feature, with the added ability to scatter rays, and a choice to scale by the diffraction limit or not.

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Frequency 1-6 The spatial frequencies (see "MTF Units") for which data is plotted.

Wavelength The wavelength number to be used in the calculation.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See the comments about vignetting factors in the discussion section below.

Field Density The field density is the number of points between zero degrees and the maximum field at which the MTF is calculated, intermediate values are interpolated. A maximum of 100 field points is allowed.

Scan Type Choose +y, +x, -y, or -x field scan direction.

Discussion This feature is nearly identical to the (diffraction) MTF vs. Field feature, except the geometric MTF is used rather than the diffraction based MTF

Geometric MTF Map

Computes the geometric modulation transfer function data as a function of field position, and displays the data over a rectangular region of field.

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

X or Y Field Width The X or Y field width in field units. This is the total width or height, not the half width or height. Field units are degrees in object space if field angle is used, otherwise field units are the same as lens units.

Frequency The spatial frequency (see "MTF Units") for which data is plotted.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Wavelength The wavelength number to be used in the calculation, or All for polychromatic GMTF.

X or Y Pixels The number of pixels at which to compute the GMTF in each respective direction. Note the size of the pixels is determined by both the number of pixels and the width of the field; the pixels are not required to be square. The GMTF is computed at the center of the pixel and the GMTF is assumed to have that value over the entire region of the pixel for display purposes.

MTF Data Choose Tangential, Sagittal, or average GMTF to be displayed.

Reference Field This control selects the field number that corresponds to the center of the map. The value zero refers to the center of the object (hx = 0, hy = 0), even if no field point is defined there.

Show As Choose grey scale or false color map as the display option.

Scatter Rays If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. See "Surface scattering settings".

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Discussion See the "Geometric MTF" for details. This feature computes the GMTF at each field point on a 2D grid. If the total number of points is large, the computation time may become quite large. See also the "FFT MTF Map".

Comment about vignetting factors

Vignetting factors determine the size and shape of the pupil as seen from different field points (see "Vignetting factors" for a full discussion). Because this analysis feature needs to trace rays at arbitrary intermediate field points where no specific vignetting factors are defined, the use of vignetting factors is not recommended. If "Remove Vignetting Factors" is checked on (the default), any defined vignetting factors will automatically be replaced with surface apertures for this computation. The surface aperture method is generally more accurate than the vignetting factor method when the pupil is overfilled with light. The resulting data may be different between the two methods. In some cases, particularly where the vignetting factors are being used to define the shape of the source beam rather than the apertures of the optics, it may be required to use the defined vignetting factors. In this case, check the "Remove Vignetting Factors" box off. Zemax will then use the closest defined field to determine the vignetting factors to use for an arbitrary field point.

RMS

Provides the Root-Mean-Square (RMS) performance data as a function or field, wavelength and defocus

RMS vs. Field

Plots RMS radial, x, and y spot radius, RMS wavefront error, or Strehl ratio as a function of field angle.

Ray Density   If the method is Gaussian quadrature, then the ray density specifies the number of radial rays to be traced. The more rays traced, the greater the accuracy, although the computation time increases. The maximum number is 20 which is sufficient for pupil aberrations up to order 40. If the method is rectangular array, then the ray density indicates the grid size. Rays outside the circular entrance pupil are ignored. See the "Discussion" section for details.

Field Density   The field density is the number of points between zero degrees and the maximum field angle specified at which the RMS/Strehl ratio is calculated, intermediate values are interpolated. A maximum of 100 field points is allowed.

Plot Scale   Sets the maximum vertical scale for the plot. Zero results in automatic scaling.

Method Select either Gaussian Quadrature (GQ) or Rectangular Array (RA). See "Comments about RMS computation methods".

Data   Selects either wavefront error, spot radius, spot x-direction, spot y-direction, or Strehl ratio. Strehl ratio is only available for monochromatic calculations, and is computed using the method described in "Strehl ratio approximation" on page 206. For a polychromatic Strehl, see the "Huygens PSF".

Refer To   Select either chief ray or centroid. For monochromatic calculations, the specified wavelength is used for reference. For polychromatic calculations, the primary wavelength is used for reference. Both reference points subtract out wavefront piston. The centroid reference mode also subtracts out the tilt of the wavefront, which yields smaller RMS values.

Orientation   Select +y, -y, +x, or -x field direction. Note the data will only be computed to the limits of the defined fields in the selected direction.

Use Dashes   Selects either solid lines or dashed lines to differentiate the various curves.

Wavelength   Select "All" to display data for each wavelength and a polychromatic computation, any one wavelength to display monochromatic data, "Poly Only" which only displays the polychromatic data, or "All With No Poly" to display all monochromatic data without the polychromatic data. Strehl ratio can only be computed for a single wavelength.

Show Diffraction Limit   If checked, then a horizontal line indicating the diffraction limited response will be drawn on the plot. For RMS radius, x, or y; the diffraction limit is assumed to be 1.22 times the working F/# on axis times the wavelength (primary wavelength if polychromatic). The change in the diffraction limit due to changes in F/# with field are ignored; a single value is used across the range of the plot. For Strehl ratio, 0.8 is used, and for RMS wavefront, 0.072 waves is used. These are all approximate indicators for convenience only; the actual meaning of "diffraction limited" may be open to interpretation.

Use Polarization   If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Remove Vignetting Factors   If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Discussion

This feature calculates the RMS error or Strehl ratio as a function of field angle for each wavelength.

Comments about RMS computation methods

Two different methods of calculation are available; either Gaussian Quadrature (GQ), or Rectangular Array (RA). For GQ, the rays traced are arranged in a radial pattern with an optimal weighting to estimate the RMS with a minimum number of rays. The method is described in G. W. Forbes, "Optical system assessment for design: numerical ray tracing in the Gaussian pupil", J. Opt. Soc. Am. A, Vol. 5, No. 11, p1943 (1988). Although the method is very efficient, the algorithm is not accurate if some of the rays are clipped due to surface apertures. To compute the RMS wavefront in systems with surface apertures requires the use of the RA method, and a larger number of rays for sufficient accuracy. For more information see "Selecting the pupil integration method" .

Comments about RMS wavefront computations

RMS wavefront error with the mean subtracted is computed as follows:

$$W_{RMS}{}^2 = \sum_{i=1}^{n} \frac{1}{w_n} \times \sum_{i=1}^{n} \left(W_n{}^2 \cdot w_n \right) - \left[\sum_{i=1}^{n} \frac{1}{W_n} \times \sum_{i=1}^{n} W_n \cdot w_n \right]^2$$

where W_n is the OPD at a given point in the pupil and w_n is the weighting at that point in the pupil. For RMS wavefront computations, Zemax always subtracts out the mean OPD. RMS wavefront, when referenced to the centroid, also requires computation of a shifted, tilted reference sphere. The raw, uncorrected OPD values do not account for the possible shift in image centroid location due to any coma that may be present. The correction

method is described in M. Rimmer, "Analysis of Perturbed Lens Systems", Applied Optics Vol. 9, No. 3, p533 (1970). This same method is used when computing RMS wavefront referenced to the centroid by the various RMS wavefront

merit function optimization operands. The method is more accurate than simple direct integration of the "raw" OPD values, which are referenced to a sphere centered on the chief ray - image surface intercept point.

RMS vs. Wavelength

Plots RMS radial, x, and y spot radius, RMS wavefront error, or Strehl ratio as a function of wavelength.

Ray Density   If the method is Gaussian quadrature, then the ray density specifies the number of radial rays to be traced. The more rays traced, the greater the accuracy, although the computation time increases. The maximum number is 20 which is sufficient for pupil aberrations up to order 40. If the method is rectangular array, then the ray density indicates the grid size. Rays outside the circular entrance pupil are ignored. See the "Discussion" section for details.

Wave Density   The wave density is the number of points between the minimum and maximum defined wavelength at which the RMS/Strehl ratio is calculated, intermediate values are interpolated. A maximum of 100 points is allowed.

Plot Scale   Sets the maximum vertical scale for the plot. Zero results in automatic scaling.

Method Select either Gaussian Quadrature (GQ) or Rectangular Array (RA). See "Comments about RMS computation methods" .

Data   Selects wavefront error, spot radius, spot x-direction, spot y-direction, or Strehl ratio.

Refer To   Select either chief ray or centroid. Both reference points subtract out wavefront piston. The centroid reference mode also subtracts out the tilt of the wavefront, which yields smaller RMS values.

Use Dashes   Selects either solid lines or dashed lines to differentiate the various curves.

Field   Select "All" to display data for all fields, select one field to display data for a single field.

Show Diffraction Limit   If checked, then a curve indicating the diffraction limited response will be drawn on the plot. For RMS radius, x, or y; the diffraction limit is assumed to be 1.22 times the working F/# on axis times the wavelength. The change in the diffraction limit due to changes in F/# with field are ignored. For Strehl ratio, 0.8 is used, and for RMS wavefront, 0.072 waves is used. These are all approximate indicators for convenience only; the actual meaning of "diffraction limited" may be open to interpretation.

Use Polarization   If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Discussion This feature calculates the RMS error or Strehl ratio as a function of wavelength for each field position. The method of calculation is identical to that described in "RMS vs. Field"; see that section for a detailed discussion. The wavelength range is determined by the minimum and maximum wavelengths defined on the wavelengths dialog box.

RMS vs. Focus

Plots RMS radial, x, and y spot radius, RMS wavefront error, or Strehl ratio as a function of focus change.
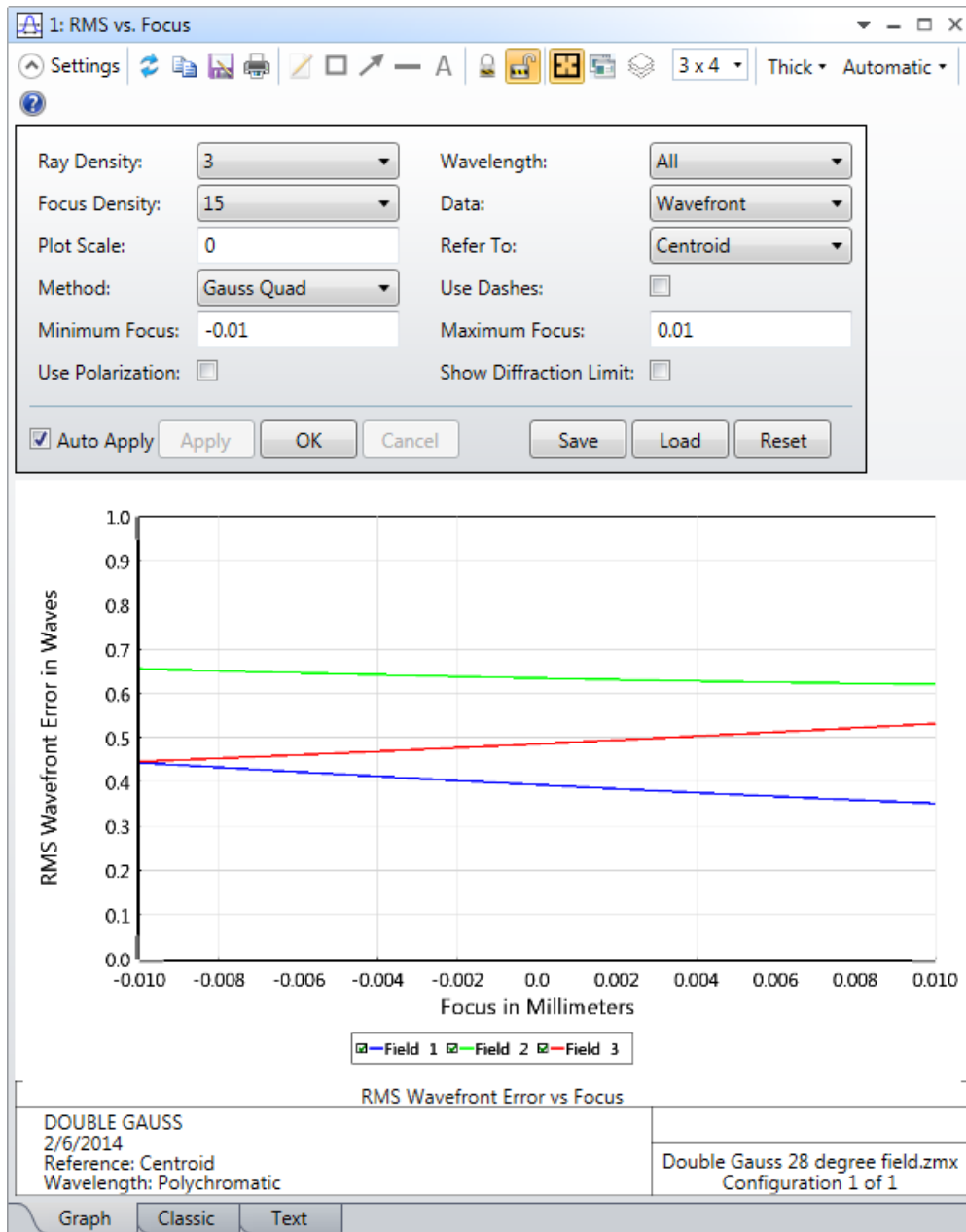
Ray Density If the method is Gaussian quadrature, then the ray density specifies the number of radial rays to be traced. The more rays traced, the greater the accuracy, although the computation time increases. The maximum number is 20 which is sufficient for pupil aberrations up to order 40. If the method is rectangular array, then the ray density indicates the grid size. Rays outside the circular entrance pupil are ignored. See the "Discussion" section for details.

Focus Density The focus density is the number of points between the minimum and maximum focus shift specified at which the RMS/Strehl ratio is calculated, intermediate values are interpolated. A maximum of 100 points is allowed.

Plot Scale Sets the maximum vertical scale for the plot. Zero results in automatic scaling.

Method Select either Gaussian Quadrature (GQ) or Rectangular Array (RA). See "Comments about RMS computation methods" .

Data   Selects either RMS wavefront error, RMS spot radius, RMS X-direction, RMS Y- direction, or Strehl ratio. Strehl ratio is only available for monochromatic calculations, and is computed using the method described in "Strehl ratio approximation" . For a polychromatic Strehl, see the "Huygens PSF".

Use Dashes   Selects either solid lines or dashed lines to differentiate the various curves.

Refer To   Select either chief ray or centroid. For monochromatic calculations, the specified wavelength is used for reference. For polychromatic calculations, the primary wavelength is used for reference. Both reference points subtract out wavefront piston. The centroid reference mode also subtracts out the tilt of the wavefront, which yields smaller RMS values.

Wavelength   Select "All" to display data for a polychromatic computation, select any one wavelength to plot monochromatic data.

Min Focus   The minimum value of the defocus to plot. For focal systems, the units are lens units. For afocal systems, the units are diopters.

Max Focus   The maximum value of the defocus to plot. For focal systems, the units are lens units. For afocal systems, the units are diopters.

Show Diffraction Limit   If checked, then a horizontal line indicating the diffraction limited response will be drawn on the plot. For RMS radius, x, or y; the diffraction limit is assumed to be 1.22 times the working F/# times the wavelength (primary wavelength if polychromatic) on axis. The change in the diffraction limit due to changes in F/# with field are ignored; a single value is used across the range of the plot. For Strehl ratio, 0.8 is used, and for RMS wavefront, 0.072 waves is used. These are all approximate indicators for convenience only; the actual meaning of "diffraction limited" may be open to interpretation.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Discussion This feature calculates the RMS error or Strehl ratio as a function of a change in focus position for each field position. The method of calculation is identical to that described in "RMS vs. Field"; see that section for a detailed discussion. Zemax adds the specified focus shift to the value of the thickness of the surface prior to the image surface. If the optical system has an odd number of mirrors, this surface normally would have a negative thickness, and therefore negative focus values move the image surface farther away from the last component. For systems with an even number of mirrors, then negative focus values move the image surface closer to the last component.

RMS Field Map

Plots RMS radial, x, and y spot radius, RMS wavefront error, or Strehl ratio as a function of field position in a rectangular map.
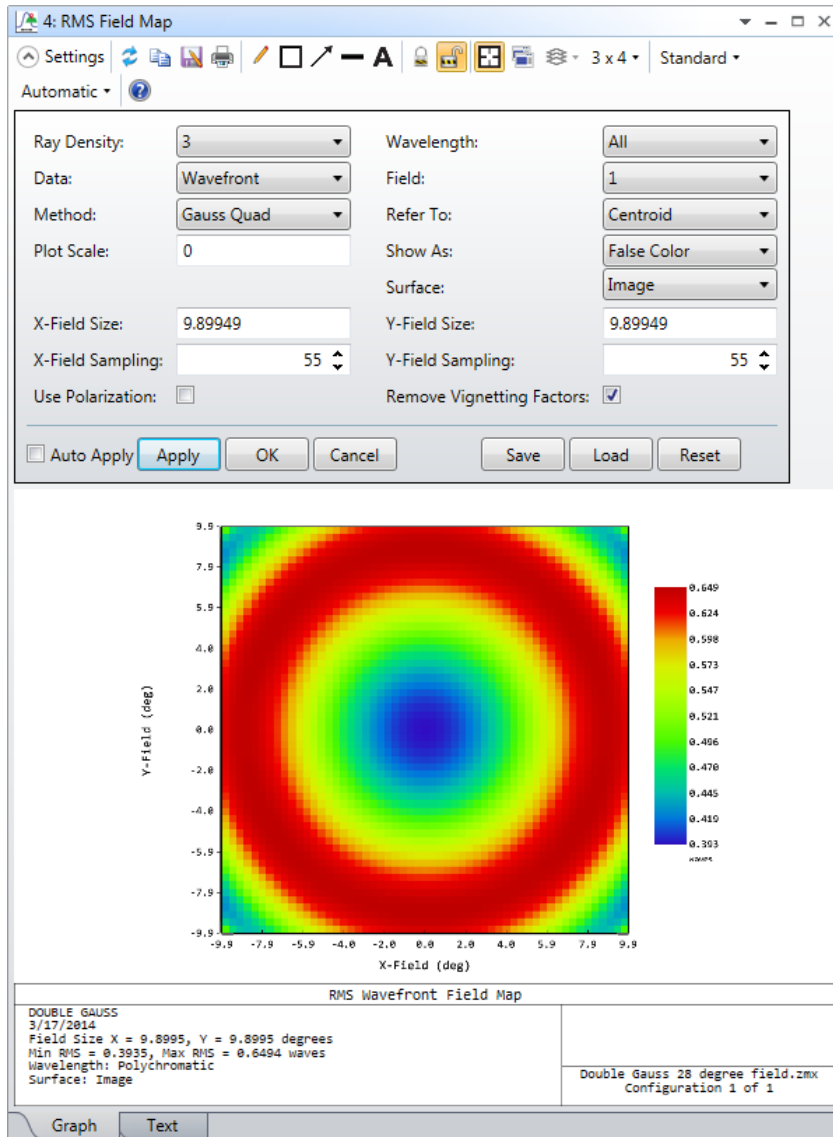
Ray Density If the method is Gaussian quadrature, then the ray density specifies the number of radial rays to be traced. The more rays traced, the greater the accuracy, although the computation time increases. The maximum number is 20 which is sufficient for pupil aberrations up to order 40. If the method is rectangular array, then the ray density indicates the grid size. Rays outside the circular entrance pupil are ignored. See the "Discussion" section for details.

Data Selects wavefront error, spot radius, spot x-direction, spot y-direction, or Strehl ratio. If the method selected is rectangular array, then a rough estimate of the PTV is also available as one of the Data options. Note that PTV depends upon just two points in the pupil, and for this reason it is difficult to obtain the exact PTV without resorting to high sampling and long computation times. Both Strehl Ratio and PTV require a single wavelength to be selected; polychromatic computations are not supported.

Method Select either Gaussian Quadrature (GQ) or Rectangular Array (RA). See "Comments about RMS computation methods".

Plot Scale Sets the maximum vertical scale for the plot. Zero results in automatic scaling.

Wavelength Select "All" to display polychromatic data, or select any one wavelength to display monochromatic data. Strehl Ratio and PTV Wavefront can only be computed for a single wavelength.

Field   Selects the reference field. The rectangular field of view will be centered on the selected field point.

Refer To   Select either chief ray or centroid. Both reference points subtract out wavefront piston. The centroid reference mode also subtracts out the tilt of the wavefront, which yields smaller RMS values.

Show As   Choose surface plot, contour map, grey scale, or false color map as the display option.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces"

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String".

X/Y Field Size   The half-field width in the X or Y directions in field units. Note this field of view is centered on the reference field selected by the "Field" setting above.

X/Y Field Sampling   The number of points in the X or Y directions used to sample the Data.

Use Polarization   If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Remove Vignetting Factors   If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Discussion This feature calculates the RMS error or Strehl ratio as a function of field position. The method of calculation is identical to that described in "RMS vs. Field"; see that section for a detailed discussion.

Enclosed Energy
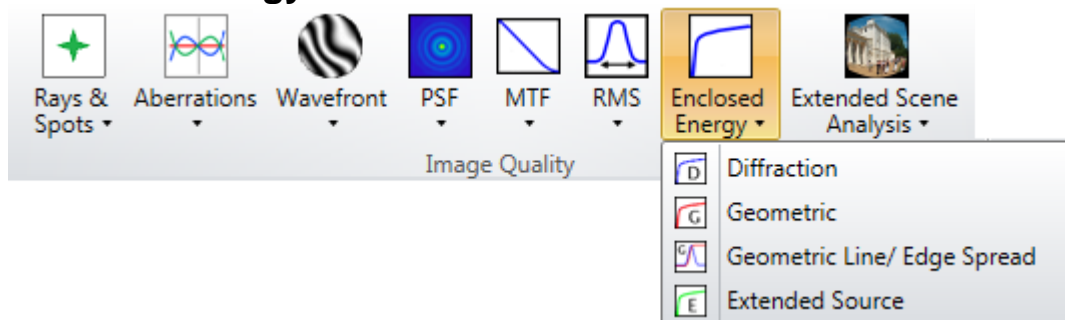
Shows the fractional energy enclosed within circular and slit regions on the specified surface

Diffraction

Encircled energy diagram. This is the percentage of total energy enclosed as a function of distance from either the chief ray or the image centroid at the image of a point object.
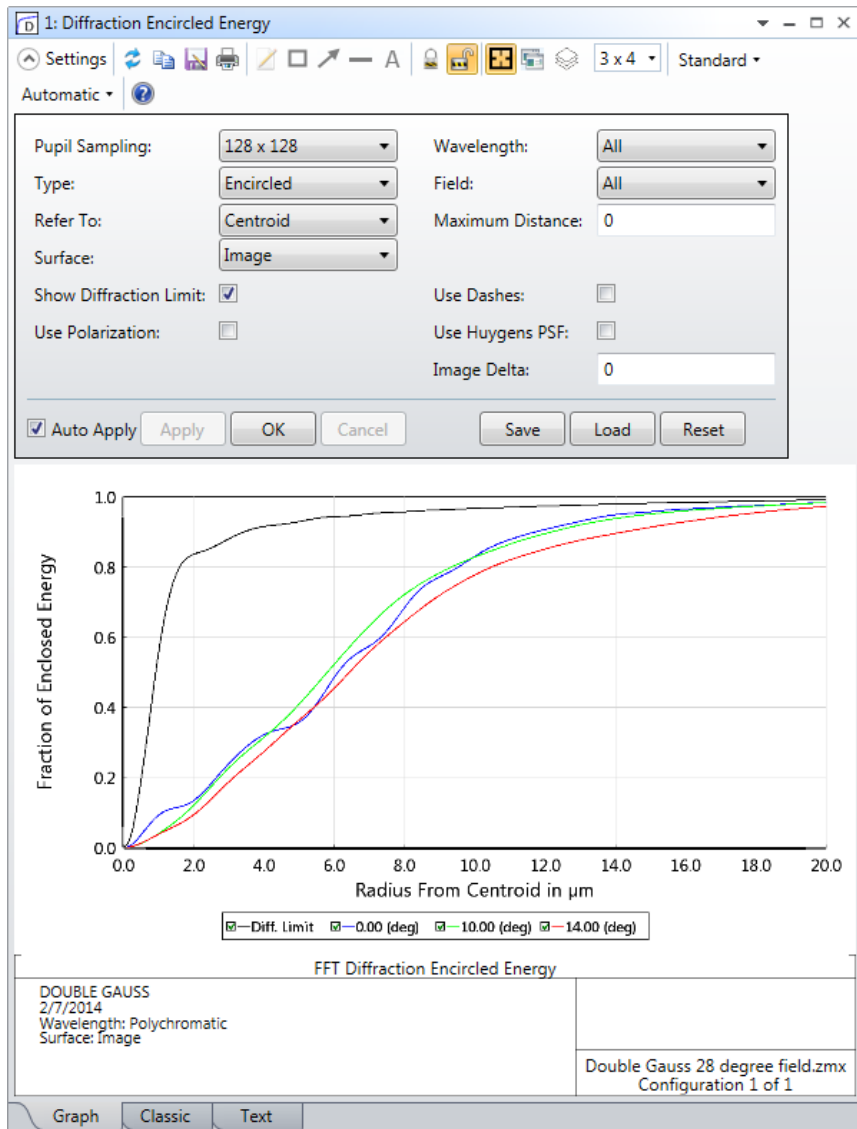
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Type The analysis type option specifies how the encircled energy is calculated; either encircled (radial), X-only, Y-only, or ensquared.

Maximum Distance This setting overrides the default scaling. The units are micrometers. To choose the default scaling option, enter zero.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves.

Refer To Select chief ray, centroid, or vertex as the reference point. Vertex refers to the coordinates (0, 0) on the image surface. This option will only return meaningful data if the diffraction image at all selected fields is within the maximum distance of the vertex. When vertex is selected, Zemax is unable to detect if the sampling is sufficient, so some care should be taken to set the sampling high enough for accurate results.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Wavelength The wavelength number to be used in the calculation.

Field   The field number for which the calculation should be performed.

Show Diffraction Limit   If checked, the diffraction limited results are computed and displayed. See the discussion below.

Use Huygens PSF   If checked, the more accurate but slower Huygens PSF method is used to compute the PSF. This option should always be used if the image surface is tilted, or if the chief ray is not close to normal to the image surface.

Image Sampling If "Use Huygens PSF" is checked, this control allows a user selectable image sampling grid. The default setting will select a grid that is 2X larger in each direction than the pupil sampling grid.

Image Delta   This control selects the point spacing on the image grid in micrometers. Use zero for the default image delta.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Discussion

See the discussion sections of the FFT and Huygens PSF. Those comments also apply to this feature.

The accuracy of the diffraction encircled energy calculation is limited by the magnitude and slope of the OPD error and the sampling density used. If the sampling density is insufficient, Zemax will issue an error message indicating that the data is inaccurate. To increase the accuracy, increase the sampling density or decrease the OPD error. If shown, the diffraction limit curve is for the aberration free response at the reference field position (see "Diffraction Limited" in the chapter "Conventions and Definitions").

For systems with low OPD error, finite energy can exist in the PSF out to long distances from the image centroid. For such systems, it can be computationally challenging to track energy accurately over both small (within the Airy radius) and large distances. When using low sampling, results can be subject to error in those cases. For such systems, higher sampling is generally required for accurate results over the full range of the PSF.

Geometric

Computes encircled energy using ray-image surface intercepts.
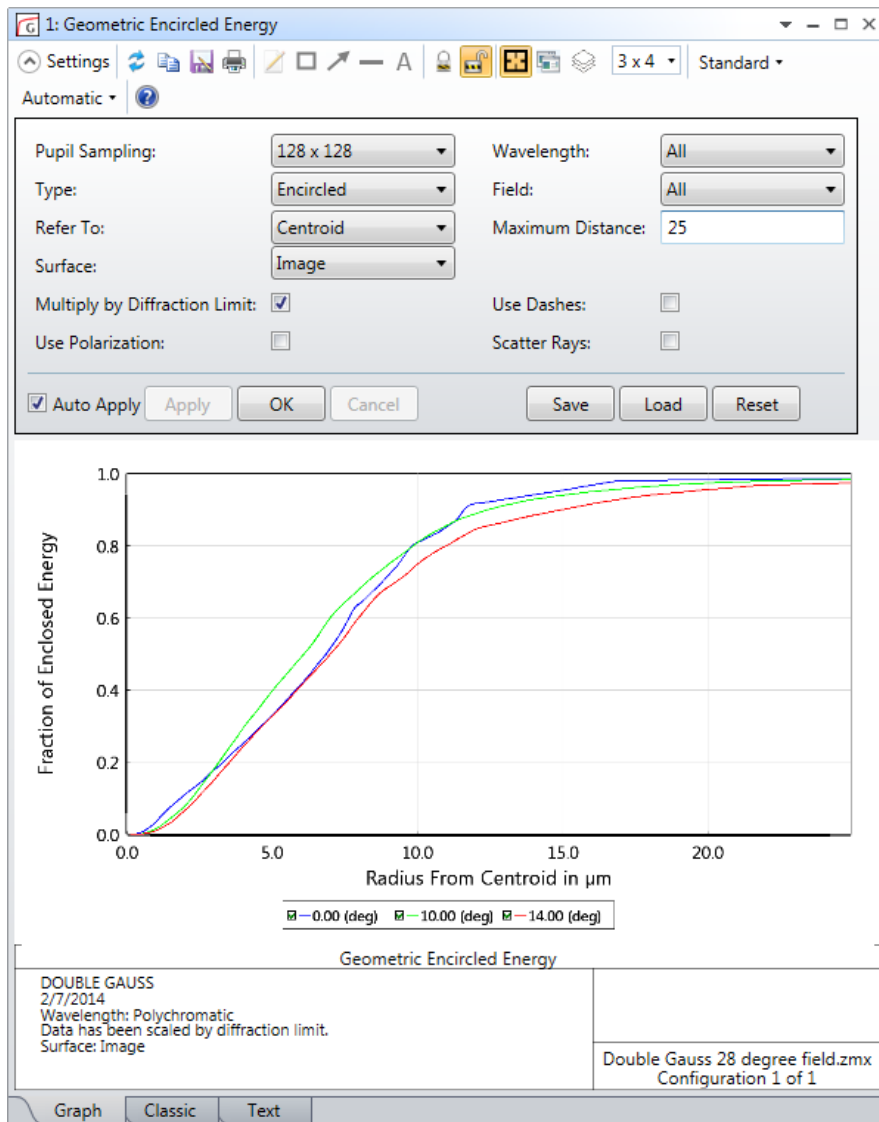
Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Type The analysis type option specifies how the encircled energy is calculated; either encircled (radial), X-only, Y-only, or ensquared.

Max Distance This setting overrides the default scaling. The units are micrometers. To choose the default scaling option, enter zero.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves.

Refer To Select either chief ray, centroid, vertex, or middle as the reference point. The middle is the coordinate upon which the smallest circle enclosing all the ray intercepts is centered.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Wavelength The wavelength number to be used in the calculation.

Field The field number for which the calculation should be performed.

Multiply by Diffraction Limit   If checked, Zemax approximates the diffraction encircled energy by scaling the geometric data by the theoretical diffraction limit curve computed for a rotationally symmetric Airy disk. The only way to compute an obscured or asymmetric pupil diffraction limit function would be to perform an exact diffraction calculation, in which case the diffraction encircled energy feature should be used instead. The diffraction limit approximation is only useful for systems with unobscured pupils, reasonably rotationally symmetric images, and modest field angles since the approximation ignores the change in F/# with field.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Scatter Rays   If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. See "Surface scattering settings".

Discussion The X- and Y-only options will compute the fraction of rays which are contained with plus or minus the specified distance from either the chief ray or the image centroid. If a scale of 10 micrometers is shown, then the region enclosed is 20 micrometers across (and infinite in the other direction). The geometric encircled energy is not a good indicator of performance if the system is close to diffraction limited.

Geometric Line/Edge Spread

Computes the geometric response to a line object and an edge object.
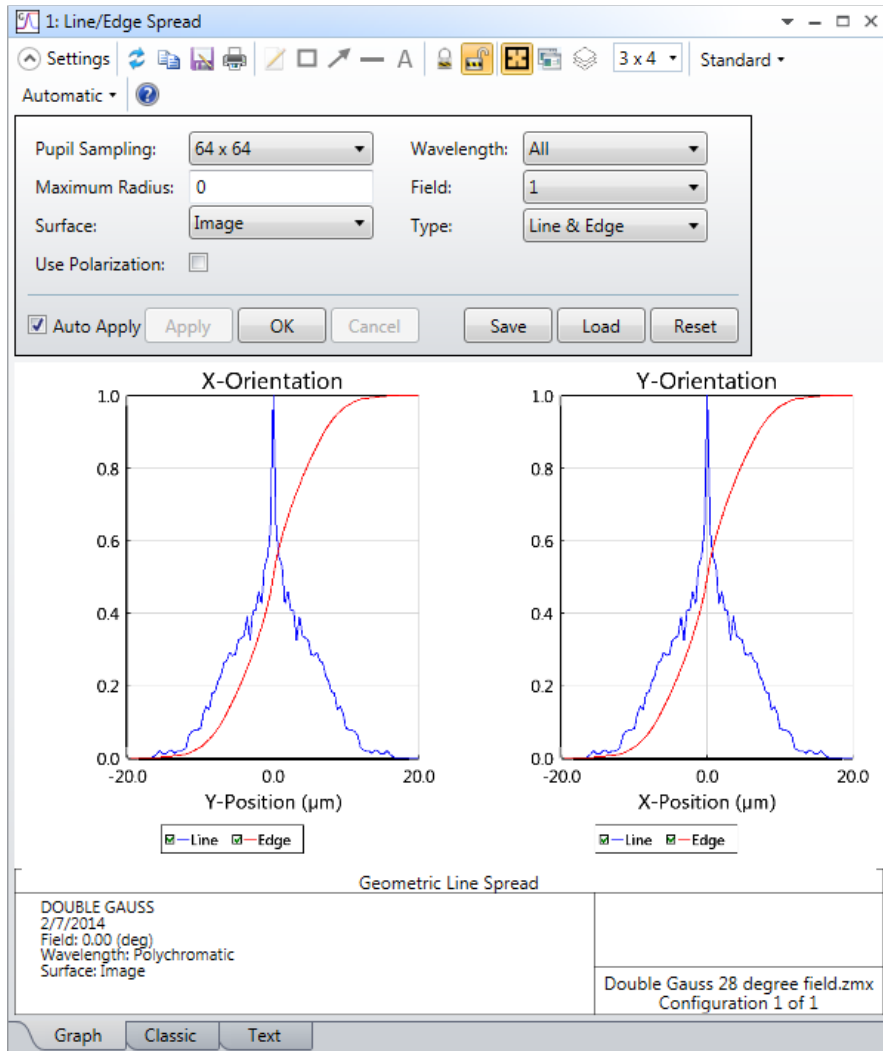
Geometric Line Spread

Sampling The size of the ray grid used to sample the pupil. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase.

Max Radius The maximum radius setting overrides the default scaling. The units are micrometers. To choose the default scaling option, enter zero.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Wavelength The wavelength number to be used in the calculation.

Field The field number to be used in the calculation.

Type The type option specifies which data is displayed on the graph; line and edge, line only, or edge only.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Discussion The line response function (or line spread function, LSF) is the cross section of the intensity pattern of the image of a line object. The edge spread function (ESF) is the cross section of the intensity pattern of the image of an edge (a semi-infinite plane). The X- or Y-orientation refers to the direction of the line or edge. The X- orientation means

the line or edge is parallel to the X-axis. The Y-orientation means the line or edge is parallel to the Y-axis. The zero coordinate is referenced to the chief ray. This is a geometric calculation. See also "FFT Line/ Edge Spread"

Extended Source

Computes encircled energy using an extended source similar to the geometric image analysis feature.
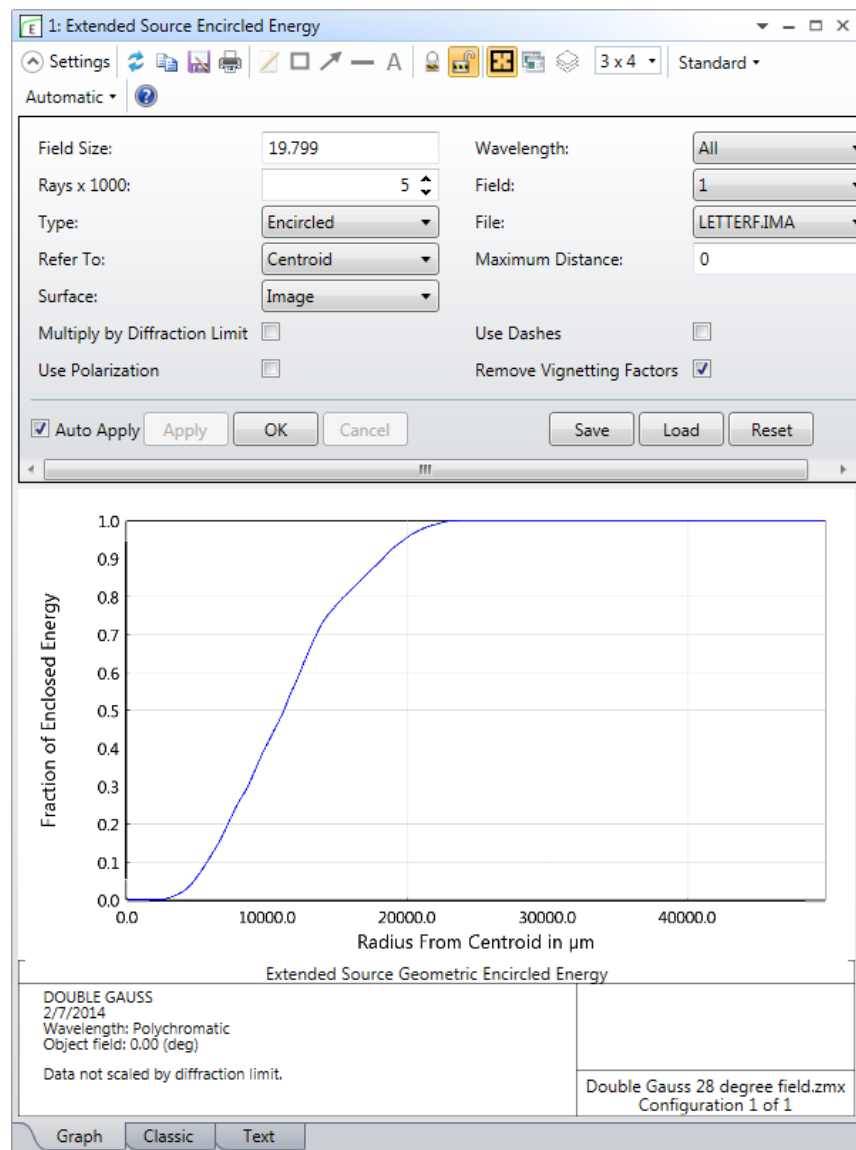
Field Size   This value defines the full width of the square image file in field coordinates, which may be either lens units or degrees, depending upon the current field definition (heights or angles, respectively).

Rays x 1000   This setting determines approximately how many rays will be traced. The number of rays traced is approximately 1000 times the specified value. The reason the number of rays is only approximate is because the distribution of rays over the pixels in the image must be uniform. For example, if there are 1500 pixels in an image file, then at least 1500 rays will be traced, even if a value of 1 is selected. The distribution of rays at each wavelength is in proportion to the wavelength weights.

Type   The analysis type option specifies how the encircled energy is calculated; either encircled (radial), X-only, Y-only, or ensquared. The X-only and Y-only options are sometimes called "enslitted" and correspond to the total fraction of energy contained within an expanding slit.

There are also options for X- or Y-distributions, which show the energy distribution in either x or y directions. These latter two options also report the geometric full width at half max. The X- or Y- distributions are the amount of energy falling on a pixel which is narrow in one direction and infinite in the other direction.

Refer To Select either chief ray, centroid, or surface vertex as the reference point.

Surface Selects the surface at which the data is to be evaluated. This is useful for evaluating intermediate images. See "Evaluating results at intermediate surfaces".

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Multiplyby Diffraction Limit If checked, Zemax approximates the diffraction encircled energy by scaling the geometric data by the theoretical diffraction limit curve. The diffraction limit curve Zemax uses is based upon the unobscured circular pupil. The only way to compute the obscured pupil diffraction limit function would be to perform the exact diffraction calculation, in which case the diffraction encircled energy feature should be used instead. The diffraction limit approximation is only useful for systems with unobscured pupils and modest field angles, since the approximation ignores the change in F/# with field. This option is only available for Type = encircled.

Wavelength The wavelength number to be used in the calculation.

Field The image file may be centered on any defined field position. This permits a small target such as a bar chart to be moved to any location in the field of view.

File The name of the .IMA image file. This file must reside in the <images> folder (see "Folders"). See the discussion section in the Geometric Image Analysis feature for a full description of the IMA file format.

Max Distance This setting overrides the default scaling. The units are micrometers. To choose the default scaling option, enter zero.

Use Dashes Selects either solid lines or dashed lines to differentiate the various curves. This setting only applies to the Classic view, which is an option if "Enable Classic View" is selected in the Graphics tab of the Project Preferences.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Discussion The X- and Y-only options will compute the fraction of rays which are contained with plus or minus the specified distance from either the chief ray or the image centroid. If a scale of 10 micrometers is shown, then the region enclosed is 20 micrometers across (and infinite in the other direction). The geometric encircled energy is not a good indicator of performance if the system is close to diffraction limited.

See the Geometric Image Analysis feature discussion for details about extended source modeling and the IMA file format.

Extended Scene Analysis
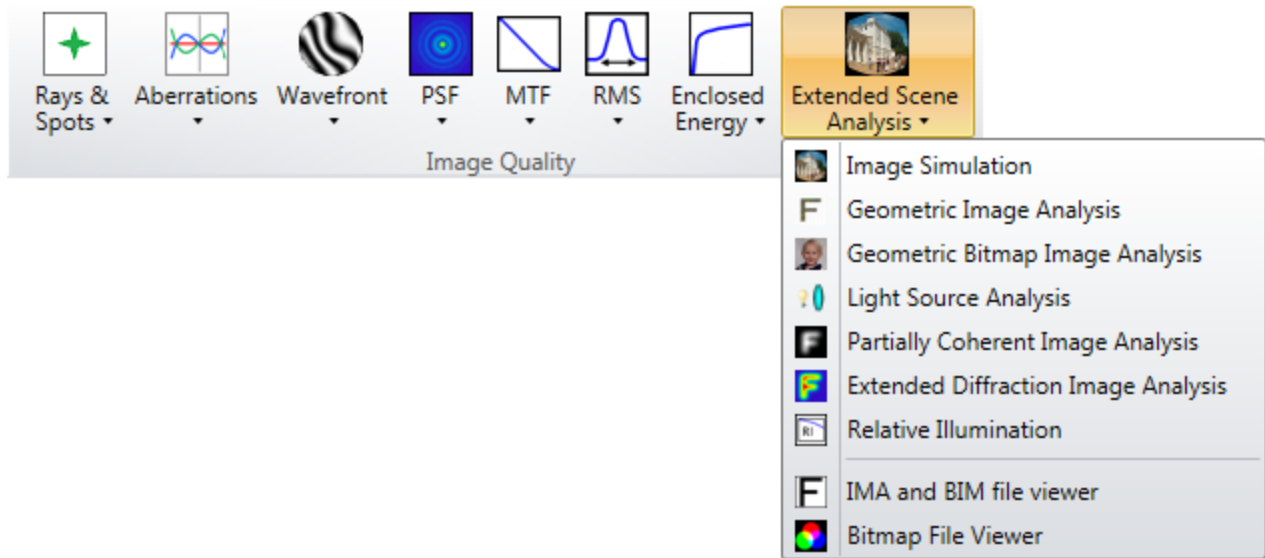
Predicts the response of the optical system to a 2D bitmap placed on the object surface

Image Simulation

This feature simulates the formation of images by convolving a source bitmap file with an array of Point Spread

Functions. The effects considered include diffraction, aberrations, distortion, relative illumination, image orientation, and polarization.
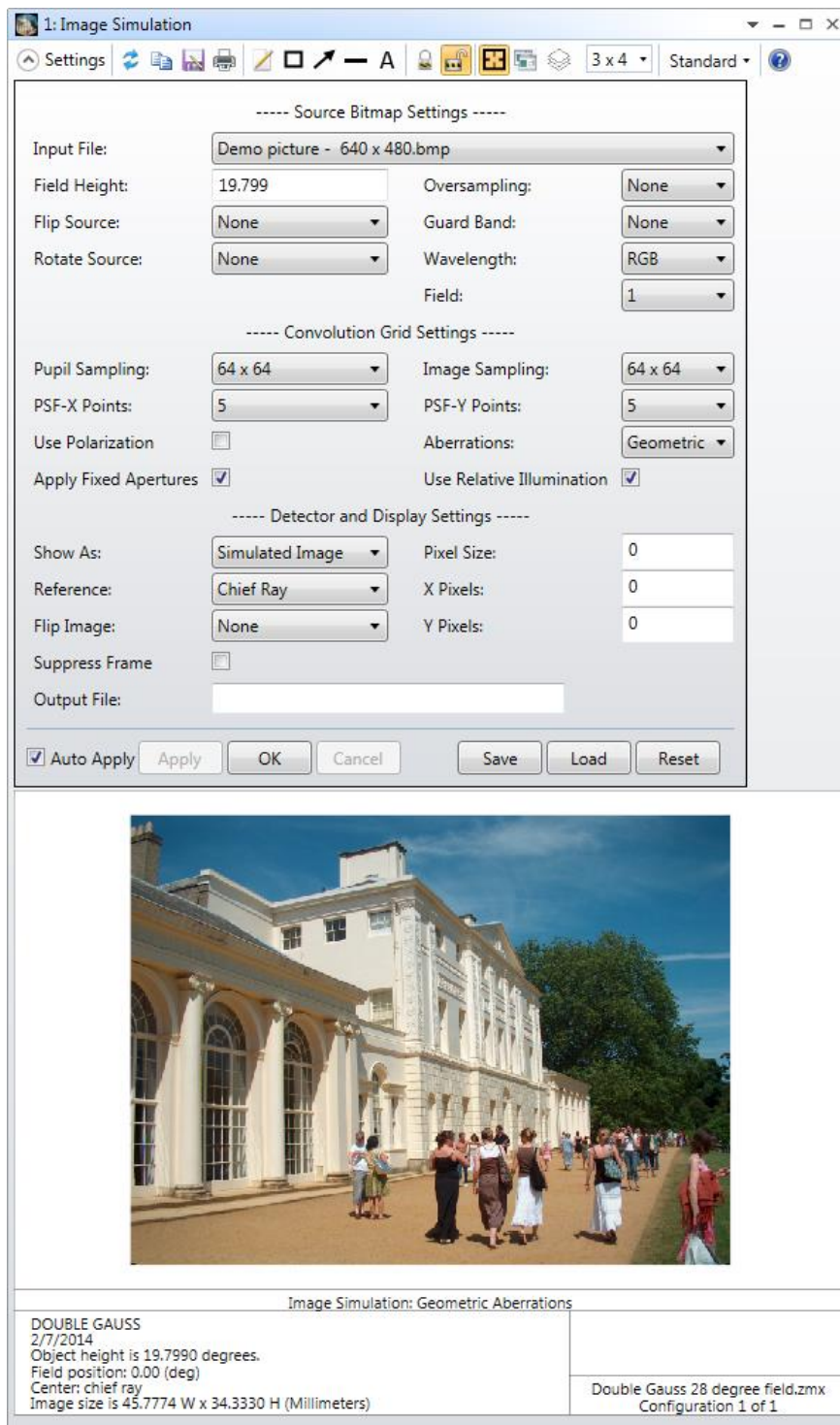
Source Bitmap Settings

Input File   The name of the file for the source bitmap. The file may be in BMP, JPG, PNG, IMA, or BIM file formats. For a description of the IMA and BIM file formats see "The IMA format" and "The BIM format" . The input files are stored in the

<images> folder (see "Folders"). IMA abd BIM files must have more than 1 but no more than 8,000 pixels in each direction, while BMP, JPG and PNG files must have more than 1 and no more than 16,000 pixels in each direction.

Field Height   This value defines the full height of the source bitmap in field coordinates, which may be either lens units or degrees, depending upon the current field definition (heights or angles, respectively). If the field type is Real Image Height, the field type is automatically changed to Paraxial Image Height for this analysis. This avoids the problem of Real Image Height masking the distortion in the image. The Field Height applies to the resulting bitmap after any oversampling, guard band, or rotation is applied.

Oversampling   Oversampling increases the pixel resolution of the source bitmap by copying one pixel into 2, 4, or more identical adjacent pixels. The purpose of this feature is to increase the number of pixels per field unit. Oversampling will be applied a factor of 2 at a time until the specified oversampling is achieved, as long as the maximum number of pixels in any direction will not exceed 16,000. This limit only applies to the oversampling feature, and not to the input file. Oversampling is applied to the image before any effects of the optical system are considered.

Flip Source   Flips the source bitmap left-right, top-bottom, or both. The source bitmap is flipped before any effects of the optical system are considered.

Guard Band   This feature increases the pixel resolution of the source bitmap by repeatedly doubling the number of pixels. This doubling only affects the analysis, while the original bitmap file remains unchanged. This results in a black "guard band" all around the original image. The purpose of this feature is to increase the number of pixels per field unit while adding a region around the desired image. This is particularly useful if the point spread function(s) are large compared to the source bitmap field size. The guard band will be applied a factor of 2 at a time until the specified size is achieved, as long as the maximum number of pixels in any direction will not exceed 16,000. This limit only applies to the guard band feature, and not to the input file.The guard band is applied to the source bitmap before any effects of the optical system are considered.

Rotate Source   Rotates the source bitmap. The rotation is applied to the source bitmap before any effects of the optical system are considered.

Wavelength   If "RGB" is selected, then 3 wavelengths will be defined, 0.606, 0.535. and 0.465 micrometers for red green, and blue, respectively; no matter what the current wavelength definitions are. If "1+2+3" is selected, then wavelengths 1, 2, and 3 as currently defined on the wavelength data box will be used. The red channel of the source bitmap will be used for wavelength 3, green channel for wavelength 2, and the blue channel for wavelength 1. The displayed image will be in RGB format no matter what wavelengths are defined using this option. For selection of a specific single wavelength, such as 6, the B channel image will be used for wavelengths 4, 7, 10, 13, etc. The G channel image will be used for wavelengths 5, 8, 11, 14, etc. The R channel image will be used for wavelengths 6, 9, 12, 15, etc.

Wavelength weighting is usually ignored. The color weighting that results in the final image depends only upon the color amplitude in the source bitmap and the relative transmission of the optical system. The exception is when using monochromatic IMA files, the selected wavelength is "1+2+3", and 3 wavelengths are defined. In this case only, the wavelength weighting is used to define the relative color weighting.

Field   The source bitmap may be centered on any defined field position. The resulting image is then centered on this field position.

Convolution Grid Settings

Pupil Sampling The grid sampling to use in the pupil space for computing the PSF grid.

Image Sampling The grid sampling to use in the source bitmap space for computing the PSF grid.

PSF X/Y Points The number of field points in the X/Y direction at which to compute the PSF. For field points in between these grid points, an interpolated PSF is used.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Apply Fixed Apertures   If checked, all surfaces with optical power that have no aperture defined are modified for this computation to have a circular aperture at the current semi-diameter value. Without this change in the aperture definition,

rays may pass the surface beyond the listed semi-diameter, especially if the Field Height exceeds the field of view defined by the field points. This leads to misleading illumination, typically at the edges of the image.

Use Relative Illumination   If checked, then the computation described in "Relative Illumination" is used to weight the rays from various points in the field of view to accurately account for the effects of exit pupil radiance and solid angle. The computation is generally more accurate, but slower, if this feature is used.

Aberrations   Choose None to ignore aberrations, Geometric to consider only ray aberrations, or Diffraction to use the Huygens PSF to model the aberrations. If Diffraction is selected, the analysis may automatically switch to Geometric if the aberrations are so severe that the diffraction PSF cannot accurately be computed. See the discussion below.

Detector and Display Settings

Show As   Choose Simulated Image, Source Bitmap to see the input bitmap (including the effects of oversampling, guard band, and rotation), or PSF Grid to see all of the PSF functions computed over the field of view. See the discussion below.

Reference   Selects the reference coordinate for the center of the plot: chief ray, surface vertex, or primary wavelength chief ray. The latter option chooses the primary wavelength chief ray even if some other wavelength is the only one selected.

Suppress Frame If checked, the frame is not drawn and the entire window is used to display the image.

X/Y Pixels   The number of pixels to use in the detected image. Use 0 for the default value, which is the number of pixels in the source bitmap.

Pixel Size   The simulated image pixel size (square). For focal systems, the units are lens units. For afocal systems, the units are cosines. Use 0 for the default value, which is computed from the magnification of the optical system for the center pixel in the source bitmap.

Flip Image   Flips the simulated image left-right, top-bottom, or both.

Output File   If a file name ending in the BMP, JPG, or PNG extension is provided, then the simulated image will be saved in the specified file and be placed in the <images> folder (see "Folders").

Discussion:

The Image Simulation algorithm consists of the following steps to compute the appearance of the image.

- The source bitmap is oversampled, rotated, and a guard band is applied, if these options are selected.

- A "grid" of Point Spread Functions (PSFs) are computed. The grid spans the field size, and describes the aberrations at selected points in the field of view defined by the bitmap and field size settings. The PSF grid also includes the effects of polarization and relative illumination.

- The PSF grid is interpolated for every pixel in the modified source bitmap. At each pixel, the effective PSFis convolved with the modified source bitmap to determine the aberrated bitmap image.

- The resulting image bitmap is then scaled and stretched to account for the detected image pixel size, geometric distortion, and lateral color aberrations.

The most important part of the algorithm is the computation of the PSF grid. The PSF X/Y Points settings determine how many PSFs are computed in each direction over the field of view. There are three different methods for modeling the aberrations using the PSF: Diffraction, Geometric, and None. Diffraction uses the Huygens PSF (for details see "Huygens PSF"). The Huygens PSF accurately models diffraction effects for almost any optical system, but is slower than the Geometric PSF, which is based upon an integration of the Spot Diagram. The Huygens PSF also cannot be accurately computed if the aberrations are severe (for this feature, the threshold is 20 times the diffraction limit). The

Image Simulation algorithm detects when the Huygens PSF cannot be computed, and will automatically switch to the Geometric PSF if required. This switching is done independently for each field and wavelength, and so imaging systems that are diffraction limited in one part of the field but severely aberrated in other parts of the field can still be accurately modeled if Diffraction is selected. If None is selected for the aberration type, the PSF grid is an array of delta functions and aberrations other than lateral color and distortion are not considered.

Relative illumination and (optionally) polarization are considered in the PSF computation. The PSF Grid is smoothly interpolated between computed field points to estimate the PSF at each pixel in the modified source bitmap. The resulting PSF is then convolved with the source bitmap to yield the aberrated image. The more PSF grid points that are used, the more accurate the simulation is, but the longer the computation time.

If the image surface is not a plane, the aberrations and distortion are all computed on the curved image surface, and the Simulated Image is projected onto the XY plane, ignoring the z coordinate of the image surface. The pixels are assumed to be square on the projected plane.

The brightness of the output image is determined by normalizing the image to have the same peak brightness as the input image.

Suggestions for use

The accuracy of the simulation is always ultimately limited by the resolution of the input image. If the resolution of the optical system is sufficient, the discrete pixel nature of the source bitmap may be apparent. This can be manifested by stair-case like edges in the simulated image. The oversampling feature can reduce these effects at the expense of longer computation time.

If the field of view is large, it is often the case that the PSFs are small compared to the area of a single pixel. In this case, a lot of computation of the PSF may yield what is effectively a delta function, with only a single pixel representing the entire PSF. The computation can be done much faster in these cases by ignoring aberrations, or at least by switching from diffraction to geometric PSF computations.

If the magnification or field of view are small enough to yield an image that is comparable in size to the PSF, or perhaps even smaller than the PSF, then the Guard Band feature is likely required. The reason is that the PSF convolution is done over the source image pixels. If the PSF is larger than the source bitmap in image space, the convolution loses the "spread" outside the original bitmap caused by the large PSF. The guard band adds a black region around the original image which can be used to show the spread due to the PSF.

By far the most important diagnostic is the "PSF Grid" under the "Show As" setting. It is always a good idea to look at the PSF grid, and confirm it shows well sampled data, before computing the Simulated Image.

Comments on using field angles

The Field Height determines the physical size of the source bitmap as seen by the optical system. For example, if a 30 x 30 pixel source bitmap is used, and the Field Height is 2.0 mm (this assumes the fields are defined in terms of object heights), then each pixel represents a 66.67 micrometer region. If the same source bitmap is later used with a system with a 40 degree full field of view (using field angles), then the Field Height can be set to 40 to cover the entire field. Each pixel will now represent a 1.33 degree square. The difficulty in using field angle for defining the object field of view is that field angle units are inherently anamorphic. X-direction angles represent a different subtended angle at a Y angle of 80 degrees than at an a Y angle of 10 degrees. If field angles are being used, and the field of view is fairly wide (more than about 40 degrees in any direction) then great care should be taken in interpreting the results for an extended object. For a precise definition of the field angles Zemax uses, see "Field angles and heights".

Comments on using paraxial and real image height

Note that if fields are defined by image height, then Field Height determines the size of the object in image space, not object space. The Field Height is always in whatever units the fields are defined in, and so when using image height as a field type the Field Height determines the source bitmap height in image space. The size of the source in object space is then determined by the Field Height divided by the magnification of the lens. When using paraxial image height, the anamorphic magnification of the system (if any) will be masked in the simulated image. When using real image height, all forms of distortion will be masked. For this reason, this feature will use paraxial image height for the field definition even if real image height is selected. However, it is recommended that neither real nor paraxial image height be used. Instead, use object height for finite conjugate systems or field angle for infinite conjugate systems, as these field types unambiguously define the size and orientation of the object in image space.

Geometric Image Analysis

The geometric image analysis feature has many applications. It can be used to model extended sources, analyze useful resolution, represent the appearance of imaged objects, and provide intuition as to image rotation. Image analysis is also useful for estimating multi-mode fiber coupling efficiency.

For a feature better suited to simulating images, see "Image Simulation"
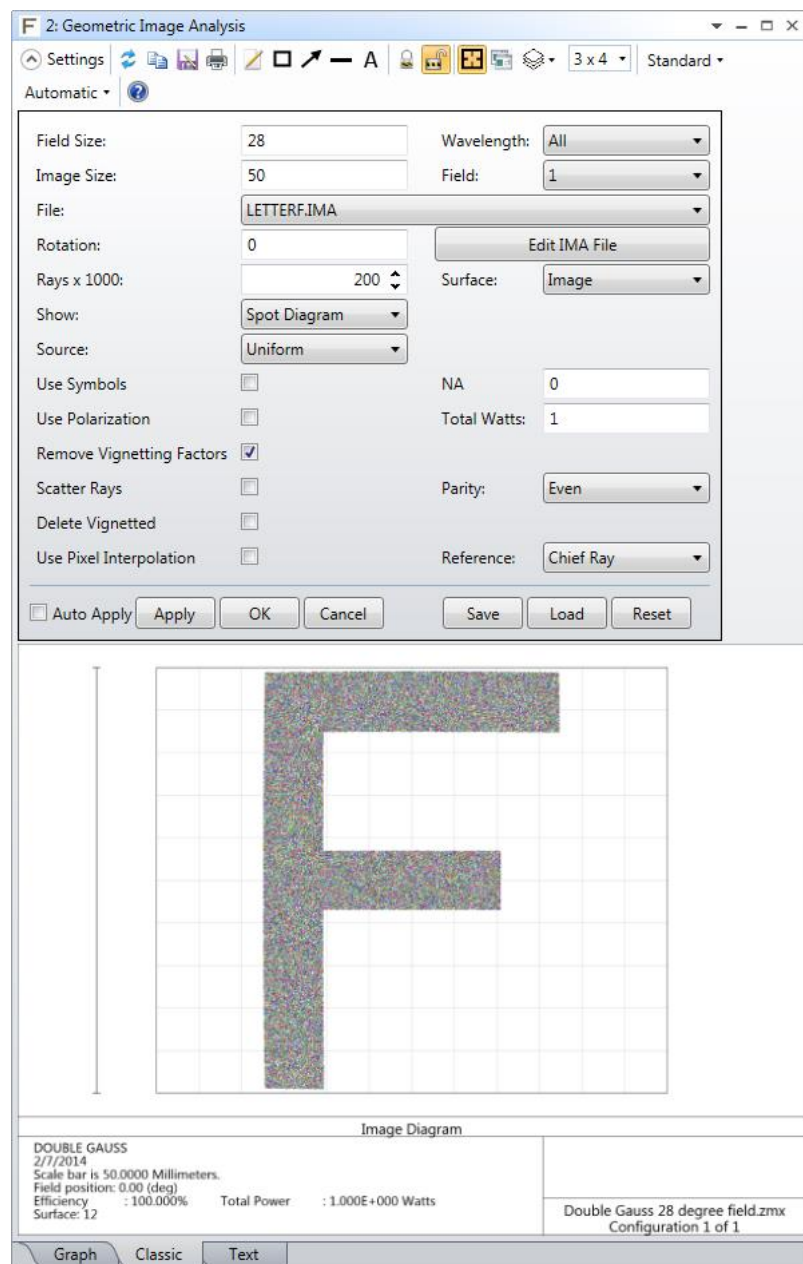
Field Size This value defines the full width of the square image file in field coordinates, which may be either lens units or degrees, depending upon the current field definition (heights or angles, respectively). Note that when using "field angle" as the field type the square IMA file is inherently distorted; see the discussion below for details.

Image Size If "Show" is selected as a spot diagram, this value sets the size of the scale bar which is superimposed on the image. It has no effect on the actual size of the image. The image size is set by the object scale and the magnification and aberrations of the system. The default may not be acceptable to see the desired portion of the image.

If "Show" is selected as any other option, this value sets the size of the detector used to capture rays. Rays landing outside of the image size are ignored, and are not included in the total detected rays, which will decrease the computed efficiency.

Parity The "Even" setting leaves the object as it would appear when viewed looking down the negative Z axis in object space. The parity can be set to "Odd" which reverses the object from top to bottom.

Rotation The rotation can be set to any angle in degrees. The algorithm actually rotates the object before tracing the rays, so this feature can be used to switch from tangential to sagittal orientation of bar targets, for example.

Rays x 1000 This setting determines approximately how many rays will be traced. The number of rays traced is approximately 1000 times the specified value. The reason the number of rays is only approximate is because the distribution of rays over the pixels in the image must be uniform. For example, if there are 1500 pixels in an image file, then at least 1500 rays will be traced, even if a value of 1 is selected. The distribution of rays at each wavelength is in proportion to the wavelength weights.

Show Choose surface plot, contour map, grey scale, false color map, spot diagram, cross x, or cross y as the display option.

Source The source may be uniform or Lambertian. The uniform setting weights rays equally. Lambertian weights rays by the cosine of the angle the ray makes with local normal of the object surface.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Scatter Rays If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. See "Surface scattering settings".

Wavelength The wavelength number to be used in the calculation.

Field The image file may be centered on any defined field position. This permits a small target such as a bar chart to be moved to any location in the field of view. The resulting image is then centered on the chief ray coordinate of this field position.

File The name of the .IMA or .BIM image file. This file must reside in the <images> folder (see "Folders"). See the discussion section for a full description of the IMA and BIM file formats.

Edit IMA File Pressing this button will invoke the Windows Notepad editor which allows modification of the currently selected IMA file. This button is disabled if the file type is BIM.

Surface The surface number at which to evaluate the rays. The default is the image surface. Other surfaces may be selected, for example, to visualize the beam footprint on an optical surface.

 # Pixels The number of pixels across the width of the selected image size. This value is not used if "spot diagram" is the method of displaying the image data.

NA The numerical aperture (NA) cut-off. If zero, this feature is ignored. If a number greater than 0 is entered, then all rays with a numerical aperture greater than the specified number are ignored.

Total Watts The total power in watts radiated by the source into the entrance pupil of the optical system. This flux is then used to normalize the detected power according to the relative pixel values and the total efficiency.

Plot Scale For false color and gray scale plots, this value defines the maximum scale range.

Use Symbols If checked, this option will draw different symbols rather than dots for each wavelength. This helps distinguish the various wavelengths. This value is only used if "spot diagram" is the method of displaying the image data.

Configuration   Select "All" to show data for all configurations at once, or select the one configuration to show, or select "Current" to show the active configuration. If "All" is selected, and both focal and afocal mode configurations are defined, the Configuration setting will automatically be reset to "Current". If "All" is selected all coordinates are referenced to the chief ray image coordinate in configuration 1, at either the primary wavelength (if all wavelengths are selected) or the selected wavelength.

Reference   Selects the reference coordinate for the center of the plot: chief ray, surface vertex, or primary wavelength chief ray. The latter option chooses the primary wavelength chief ray even if some other wavelength is the only one selected.

Remove Vignetting Factors   If checked, vignetting factors are automatically removed. See "Comment about vignetting factors"

Delete Vignetted   If checked, rays are not drawn if they will be vignetted by any surface.

Use Pixel Interpolation   If checked, ray energy is distributed among adjacent pixels according to the distance from where the ray lands to the center of the pixel. This yields smoother, more accurate results for images where the pixels are small compared to the spot size. If unchecked, all the energy for a ray is placed only in the pixel struck. Pixel Interpolation should not be used if the pixels are large compared to the spot size. This control has no affect if "Show" is set to "spot diagram".

Save As BIM File   If a file name ending in the BIM extension is provided, and "Show" is not set to "Spot Diagram", then the output image will be saved in the specified file and be placed in the <images> folder (see "Folders").

Discussion

Zemax supports three different file formats. Two of these formats end in the IMA extension, one in the BIM extension

The IMA format

There are two different IMA file formats, one text and one binary. Whichever file format is used, the file must end in the extension IMA. Zemax will distinguish between the two types of file formats automatically.

The text image file is a text file which ends in the extension .IMA. At the top of the file is one number which indicates the size of the file in pixels. The remaining rows and columns contain the pixel data, with one character to each pixel. All IMA files must be square, with n x n pixels defined. For example, a 7 x 7 representation of the letter "F" could be described by the following IMA file:

7

0111110

0100000

0100000

0111100

0100000

0100000

0100000

Note that the single entry "7" starts the file, and it is followed by a carriage return. Then there are 7 rows of 7 columns each, and each row is terminated with a carriage return. The columns are not separated by a space or any other character. The image file must be square. Zemax will attempt to allocate enough memory to hold the image file and will report an error if there is not enough memory.

The "intensity" at each pixel can be any digit between 0 and 9. The number of rays each pixel will generate is proportional to this value. Pixels with a value of 0 do not radiate any rays.

The binary IMA file format is more complicated than the text format, and binary IMA files cannot be edited with a text editor. However, the binary IMA files are dramatically more powerful. Each pixel in the binary IMA file is represented by an unsigned byte, which means there are 256 "gray-scale" levels of intensity. Furthermore, each wavelength can be assigned a separate pixel map. Therefore, very realistic photograph like extended sources can be modeled.

The binary IMA file format requires 3 16-bit header values. The first 16-bit value is a signed integer that must be equal to zero. The second 16-bit signed integer is the width of the pixel map in pixels, which can be any number from 1 to 8000. The third 16-bit signed integer is the number of pixel maps, which correspond to the number of colors (or wavelengths) represented in the file.

For example, a 3-color binary pixel map of a 50 by 50 image would have 6 bytes of header (0, 50, and 3), followed by 2500 bytes for color 1, then 2500 bytes for color 2, then 2500 bytes for color 3, for a total of 7506 bytes. The data for each color is stored by columns for each row (the column index changes faster than the row index).

The BIM format

The drawback to the IMA format is that a maximum of 256 grey scale levels are supported. The BIM format is a binary double precision floating point file format which effectively makes the number of grey scales many trillions. The BIM format consists of the following binary values:

1 32 bit integer representing the number of x pixels, nx.

1 32 bit integer representing the number of y pixels, ny.

followed by n*n 64 bit double precision floating point values representing the relative intensity. The first pixel is the bottom left corner, and the remaining pixels are listed by rows along x.

Currently, the nx and ny values must be identical or an error message will be issued.

Comments on using field angles

The comments regarding field angles for the Image Simulation also apply to this feature. For a full discussion, see "Comments on using field angles" .

Comments on using paraxial and real image height

The comments regarding paraxial and real image height for the Image Simulation also apply to this feature. For a full discussion, see "Comments on using paraxial and real image height" .

How rays are chosen for analysis

The rays generated by each pixel are chosen randomly from coordinates within the pixel cell. The entrance pupil coordinates are also chosen randomly for each ray. The distribution of rays is uniform over the pixel and over the circular paraxial entrance pupil (if ray aiming is used, then there may be some pupil distortion). For the text IMA files, the number of rays generated by each pixel is equal to the pixel intensity times the number of wavelengths times the ray density. The wavelength used for each ray is selected randomly in proportion to the wavelength weights. For the binary IMA files, the number of rays generated from each pixel is proportional to the ray density times the fractional intensity relative to 256.

By separating the form of the object from the scale, the same image file can be used for many applications. For example, the sample image file "letterf.ima" contains a 7x7 grid of pixels defining the capital letter F. The object scale can be set to 1 mm, then 0.1 mm, then 0.01 mm to get a feel for how small a character F the optical system can resolve, without the need to change the IMA file.

Note that if fields are defined by image height, then field size determines the size of the object in image space, not object space. The field size is always in whatever units the fields are defined in, and so for image height the field size determines image height. The size of the object is then determined by the field size divided by the magnification of the lens.

The choice of field position also permits great flexibility in analyzing image quality. For example, the letter F image file can be tested at several field points to see if the resolution is strongly affected by field aberrations. The object scale is set to the height of the letter, but the image will be centered about the chief ray intercept of the selected field point.

The source is by default a uniform radiator of rays. Uniform here means uniform in the entrance pupil. All rays generated fall within the entrance pupil, and they are all weighted equally. Since ray wavelengths are selected randomly in proportion to the wavelength weights, no explicit wavelength weighting is required. The uniform setting is usually preferred for large object distance systems with small fields of view. The source may also be defined to be Lambertian, which weights all rays by a cosine factor. Once the input ray weight is determined, any pupil apodization (see "Apodization Type") that is defined will further modify the input ray weight.

The percent efficiency is defined by

$$\%E = \frac{\sum W_i}{\sum W_j}$$

where the sum i is over all rays which were unvignetted, and the sum over j is over all rays which were launched. The efficiency calculation considers vignetting, source distribution, wavelength weights, and reflection and transmission losses in the optical system if the "Use Polarization" checkbox is selected. If "spot diagram" is selected for the "show" option, then the percent efficiency includes all unvignetted rays. The other display options vignette rays that are beyond the extent of the detector size. Therefore, the percent efficiency will be different between spot diagram displays and other displays if rays fall outside the region of the detector.

Calculating efficiency of multi-mode fibers

Zemax has an algorithm for accurately computing fiber coupling into single-mode fibers; for details see "Fiber Coupling Efficiency". Also see "Computing Fiber Coupling".

To estimate the coupling efficiency for multi-mode fibers, a geometric approach may be used. Place a circular aperture at or just before the image surface with the appropriate maximum radial aperture representing the core size. Then set the NA (see the table above) to the maximum acceptance NA of the fiber. The percent efficiency will then be calculated by summing all the rays that pass the core aperture within the specified NA. The NA of a typical multi-mode fiber with an inner core of index n_i and an outer cladding of index n_o is given by

$$NA = \sqrt{n_i{}^2 - n_o{}^2}$$

Text output

Selecting the "Text" option on the image analysis window menu bar will generate and display a text file listing the ray data. If the "Show" option is set to "Spot Diagram", the file will have 9 columns. The first column is the sequential ray number. The second and third columns are the x and y field coordinates (either degrees or object height). The fourth and fifth columns are the normalized pupil coordinates, Px and Py. The sixth column is the integer wavelength number. The seventh column is the weight of the ray, which depends upon the source properties. The eighth and ninth columns are the image coordinates in lens units, relative to the reference ray.

If the "Show" option is not set to "Spot Diagram", then the text display will list the weighted ray count in each pixel. The pixel listing starts at the -x, -y (bottom left) corner pixel, and proceeds up the y columns. Use the "Escape" key to terminate a lengthy image analysis computation.

Geometric Bitmap Image Analysis

The feature "Image Simulation" is generally superior to this feature, unless the simulated image is desired on an intermediate surface away from the image.

This feature creates an RGB color image based upon ray tracing data using an RGB bitmap file as the source. This feature has many applications. It can be used to model extended sources, analyze useful resolution, display distortion, represent the appearance of imaged objects, provide intuition as to image rotation, display beam footprints, and show surface plots of illumination on any surface, to name just a few.

This feature is based strictly upon geometrical ray tracing. The bitmap image analysis feature uses standard BMP, JPG, and PNG files as the source image, see the discussion for details

Field Y Size This value defines the full y direction size of the source bitmap in field coordinates, which may be either lens units or degrees, depending upon the current field definition (heights or angles, respectively).

Parity The "Even" setting leaves the object as it would appear when viewed looking down the negative Z axis in object space. The parity can be set to "Odd" which reverses the object from top to bottom.

Rotation The rotation can be set to any angle in degrees. The algorithm actually rotates the object before tracing the rays, so this feature can be used to switch from tangential to sagittal orientation of bar targets, for example.

Rays/Pixel This setting determines how many rays per source image pixel per color channel will be traced. Selecting 10 rays/pixel on a 640 x 480 3-color image will in general trace 9,216,000 rays.

X-Pixels The number of pixels across the X direction on the detector.

Y-Pixels The number of pixels across the Y direction on the detector.

X-Pixel Size The size in lens units of each detector pixel measured in the X direction.

Y-Pixel Size The size in lens units of each detector pixel measured in the Y direction.

Source The source may be uniform or Lambertian. The uniform setting weights rays equally. Lambertian weights rays by the cosine of the angle the ray makes with local normal of the object surface.

Normalize The image may be normalized by either the peak pixel intensity or by the average pixel intensity. Using the peak pixel intensity scales all pixels equally so the pixel with the highest intensity determines the overall brightness of the image. This method generally produces fairly dark images if the signal-to-noise ratio is low. Using the average pixel intensity scales all pixels equally until the average brightness equals the original bitmap image brightness. The disadvantage to this method is that some pixels become oversaturated, and are thus clipped at the maximum brightness that can be displayed.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Grey Scale If checked, the RGB intensities will be averaged at each detector pixel to yield a grey scale detected image. Rays will still be traced according to the relative RGB intensities of the source bitmap, but all color information will be lost when the detected image is displayed.

Wavelength If "RGB" is selected, then 3 wavelengths will be defined: 0.656, 0.587. and 0.486 micrometers for red green, and blue, respectively; no matter what the current wavelength definitions are. If "1+2+3" is selected, then wavelengths 1, 2, and 3 as currently defined in the wavelength data box will be used. The red channel of the source bitmap will be used for wavelength 3, the green channel for wavelength 2, and the blue channel for wavelength 1. The displayed image will be in RGB format no matter what wavelengths are defined using this option. Also note that if using this option in a system with less than 3 wavelengths defined in the wavelength data box, then the highest defined wavelength will be used to "fill in" the remaining channels. For selection of a specific wavelength, such as 1, 2, 3, etc... the B, G, or R channel image will be used; for wavelengths higher than 3 the R channel is always used.

Field The source image may be centered on any defined field position. This permits a small target such as a bar chart to be moved to any location in the field of view. The resulting image is then centered on the chief ray coordinate of this field position.

Input The name of the BMP, JPG, or PNG source image file. This file must reside in the <images> folder (see "Folders").

Surface The surface number at which to evaluate the rays.

Show Source Bitmap If checked, then the source bitmap will be drawn and no rays will be traced. The number of rays, pixels, and pixel sizes are ignored when drawing the source bitmap. This feature is used to check the appearance of the source bitmap and to verify Zemax properly reads the bitmap file.

Output The name of the BMP, JPG, or PNG file to write the detected bitmap to. The detected bitmap size is determined by the number of x and y pixels defined; but the pixels size must be the same in x and y for the aspect ratio to be correct in the output bitmap file. The file name must end in a BMP, JPG, or PNG extension, with no path name supplied. This file will be created or overwritten without warning and will be placed in the <images> folder (see "Folders").

Reference Selects the reference coordinate for the center of the plot, either the chief ray or the surface vertex.

Delete Vignetted If checked, rays are not drawn if they will be vignetted by any surface.

Suppress Frame If checked, the frame is not drawn and the entire window is used to display the image.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Apply Fixed Apertures If checked, all surfaces with optical power that have no aperture defined are modified to have a circular aperture at the current semi-diameter value. Without this change in the aperture definition, rays may pass the surface beyond the listed semi-diameter, especially if the Field Y Size exceeds the field of view defined by the field points. This leads to misleading illumination, typically at the edges of the image.

Discussion

See also "Geometric Image Analysis", as this feature is very similar.

Comments on field units

The Field Y Height determines the physical size of the source file as seen by the optical system. For example, if a 50H x 100W pixel source file is used, and the Field Y Height is 2.0 mm (this assumes the fields are defined in terms of object or image heights), then each pixel represents a 0.040mm x 0.040mm region, and the source bitmap covers a 2.0mm high x 4.0 mm wide area.

Note that if fields are defined by image height, then Field Y Height determines the size of the object in image space, not object space. The Field Y Height is always in whatever units the fields are defined in, and so when using image height as a field type the Field Y Height determines the source bitmap height in image space. The size of the source in object space is then determined by the Field Y Height divided by the magnification of the lens. When using paraxial image height, the anamorphic magnification of the system (if any) will be masked in the simulated image. When using real image height, all forms of distortion will be masked. For this reason, this feature will automatically switch to paraxial image height for the field definition if real image height is selected. However, it is recommended that neither real nor paraxial image height be used. Instead, use object height or field angle, as these field types unambiguously define the size and orientation of the object in image space.

Selection and tracing of rays

The rays generated by each pixel are chosen randomly from coordinates within the source pixel cell. The entrance pupil coordinates are also chosen randomly for each ray. The distribution of rays is uniform over the pixel, the ray distribution over the entrance pupil is uniform unless pupil apodization is defined.

Once a ray is generated, it is traced through to the selected surface. If the ray is vignetted or an error occurs, the ray is ignored. Otherwise, the pixel on the receiving detector that the ray struck is determined, and the intensity of the ray is added to the detector's bin count in the appropriate color channel.

After all the rays are traced, an RGB image is created from the normalized counts in each detector pixel. The percent efficiency is defined by

$$\%E = \frac{\sum W_i}{\sum W_j}$$

where the sum i is over all rays which were unvignetted, and the sum over j is over all rays which were launched. The efficiency calculation also includes consideration of reflection and transmission losses in the optical system if the "Use Polarization" checkbox is selected, as well as ray errors. Rays that are beyond the extent of the detector are considered to be vignetted.

This feature automatically divides the computation over all available processors for maximum execution speed. Any symmetry in the lens is also used to speed up the ray tracing computation. Use the "Escape" key to terminate a lengthy image analysis computation.

Light Source Analysis

This feature is only available in the Premium edition of Zemax.

This feature creates an image based upon ray tracing a list of rays as the source. The rays are defined by either a DAT or a SDF Source File (only binary format ray files are supported), as used by the NSC Source File object described in "Source File.

This feature is based strictly upon geometrical ray tracing. See the discussion for details.
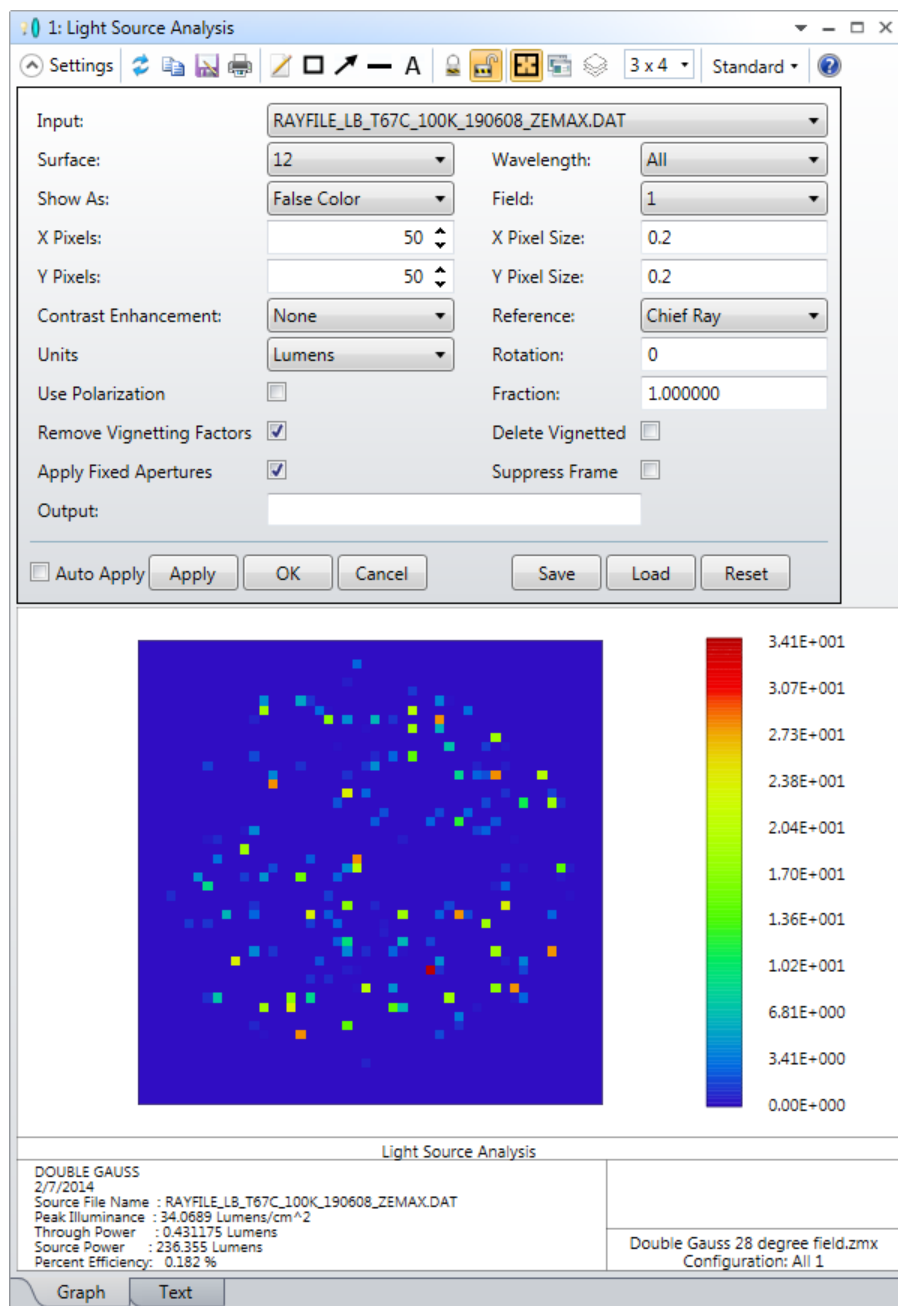
Input The name of the DAT or SDF source file. For a complete description of this file format see "Source File".

Surface The surface at which to evaluate the rays.

Wavelength The wavelength used to trace the rays for DAT files. The rays may either be monochromatic, or may be based upon the defined wavelengths and weighting. If a SDF file is used, the wavelengths specified in the SDF file are always used, regardless of the system wavelength settings.

Show As Choose Grey Scale or False Color for the intensity display. If Lumens are selected for the units, then True Color is also a supported option.

Field The field position to place the source coordinate system at. It is the object point location on the object surface that define the origin of the rays in the DAT or SDF input file.

X/Y Pixels   The number of pixels across the X/Y direction on the detector.

X/Y Pixel Size   The size in lens units of each detector pixel measured in the X/Y direction.

Contrast Enhancement   By default, no scaling of the data is done. If Contrast Enhancement, a non-linear weighting function is applied to the pixel data to exaggerate the less intense pixels. This can be useful to bring out fine detail in the image, however, the visual appearance may be misleading since dim regions will be brighter in the image than they would really be.

Reference   This control defines the origin of the detector on the selected analysis surface. Either the chief ray at the primary wavelength for the selected field, or the surface vertex may be used to define the center of the detector.

Rotation   This control rotates the entire source ray list about the +Z axis prior to tracing the rays. The angle is in degrees, and the rotation is counterclockwise in the XY plane.

Units   Choose either Watts for radiometric analysis, or Lumens for photometric analysis. Note True Color display (for Show As) is only supported when using Lumens as the units.

Fraction   Only that fraction of rays from the file will be traced. If the fraction is less than 1.0, the rays which are traced from the file are randomly selected, though for a given fraction the randomly selected rays will be identical from one analysis to the next. This feature is useful for tracing a smaller number of rays than is defined in the file.

Use Polarization If checked, polarization ray tracing will be used to consider transmission. Because no polarization data is stored in the source ray file, the polarization will be assigned randomly.

Delete Vignetted If checked, rays are not drawn if they will be vignetted by any surface.

Suppress Frame If checked, the frame is not drawn and the entire window is used to display the image.

Remove Vignetting Factors   If checked, vignetting factors are automatically removed. See "Comment about vignetting factors".

Apply Fixed Apertures   If checked, all surfaces with optical power that have no aperture defined are modified to have a circular aperture at the current semi-diameter value. Without this change in the aperture definition, rays may pass the surface beyond the listed semi-diameter, especially if the Field Y Size exceeds the field of view defined by the field points. This leads to misleading illumination, typically at the edges of the image.

Output The name of the BMP, JPG, or PNG file to write the detected bitmap to. The detected bitmap size is determined by the number of x and y pixels defined; but the pixels size must be the same in x and y for the aspect ratio to be correct in the output bitmap file. The file name must end in a BMP, JPG, or PNG extension, with no path name supplied. This file will be created or overwritten without warning and will be placed in the <images> folder (see "Folders").

Discussion

This feature traces rays representing a complex light source through a sequential optical system. The light source is modeled using either a flux-only DAT file, or a full spectrum SDF file, as described in the NSC Chapter under "Source File". Only binary format ray files are supported.

NSC Source Files may contain rays that start at any arbitrary point in space, and may travel in any direction, and be defined at any wavelength. This generality is not fully supported in the sequential ray tracing paradigm, so a few assumptions are needed.

First, the specified field point on the object surface is assumed to be the origin for all rays in the ray file, with the +Z axis of the source coordinate system parallel to the +Z axis of the object surface. The source rays may be rotated about this origin using the Rotation angle setting. Because the source rays may have a starting Z coordinate that is not zero, rays may start either behind or in front of the actual object surface. It is assumed that the first surface is far enough away from the object surface so that no rays start inside of the first or subsequent sequential surfaces.

Second, rays that travel toward the optical system may be traced beyond the limits of the entrance pupil. As long as the ray is traceable, the ray will continue through the sequential system, even if the ray is beyond the normal acceptance region of the entrance pupil. This is why it is important to use the "Apply Fixed Apertures" feature; otherwise, rays that can be traced beyond the sequentially determined aperture size may be traced all the way to the detector.

Use the "Escape" key to terminate a lengthy image analysis computation.

Partially Coherent Image Analysis

This feature considers diffraction and aberrations of the optical system, as well as partial coherence of the illumination, to compute the image appearance. This method accounts for the finite pass band and other diffraction related effects real optical systems have on image formation. This feature can compute the coherent, incoherent, or partially coherent diffraction image of scene defined by a bitmap image. For purely incoherent analysis "Image Simulation" is generally superior.

The image analysis feature uses IMA/BIM files to describe the object to be imaged. See "The IMA format" and "The BIM format".
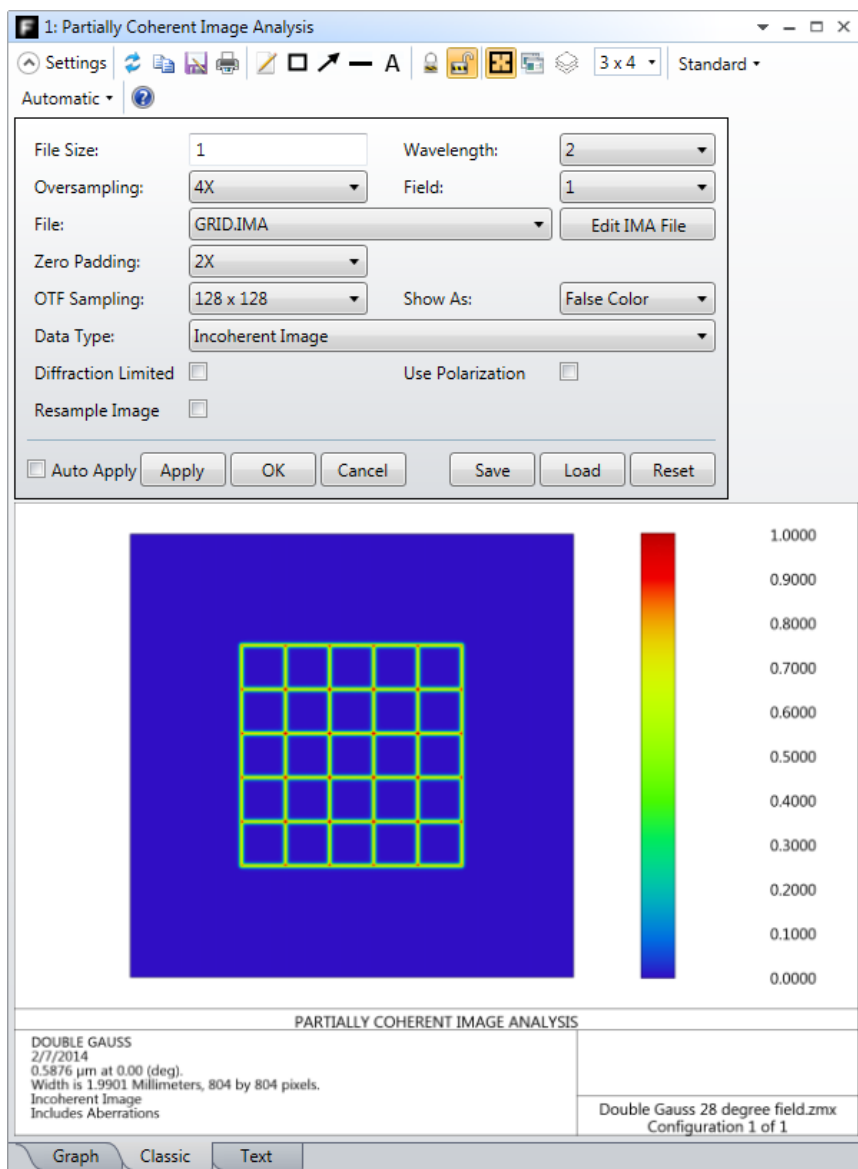
File Size The full width in lens units of the region defined by the file measured in image space. Note IMA and BIM files are always square. This feature also supports a subset of ZBF files, see the discussion for details.

Oversampling   Sets the factor by which the file pixels are oversampled. This increases the effective resolution of the file without the need to define a new file. If the original file has an odd number of pixels, oversampling will make the number of pixels even because the oversampling values are all even.

Zero Padding   Determines the actual size of the region to compute the diffraction image of by adding zero intensity values around the file pixels. This increases the size of the displayed diffraction image without changing the size of the unaberrated image; this allows study of the energy diffracted well away from the perfect image location.

To keep the file data centered, zero padding will yield an odd number of pixels if the original IMA file has an odd number of pixels. The number of pixels will be 2*n-1, where n is the original number of pixels. If the original file has an odd number of pixels, the modified image will have 2*n pixels.

Zero Padding only affects IMA and ZBF, and not BIM format files.

OTF Sampling   The grid size of the sampling in the pupil; larger grids yield a more accurate representation of the system OTF. This has no affect on the size of the diffracted image, just the accuracy of the predicted frequency response.

Show As   Choose surface plot, contour map, grey scale, false color, cross section x, or cross section y as the display option. For information on cross sections, see "Computing MTF using Cross Section X/Y".

Data Type   Choose either the incoherent image, coherent image, raw image, incoherent transfer function, coherent transfer function, or the transform of the raw image.

For partial coherence computations, choose partially coherent image (either mostly coherent or mostly incoherent), or partially coherent PSF or gamma. For a full discussion on partially coherent image computations, see "Computing partially coherent images".

See the discussion below for information about the coherent transfer function and the limitations of this computation.

Gamma   The form of the complex degree of coherence function.

Alpha   The scaling parameter used in the complex degree of coherence function.

Fraction   The fraction of encircled energy in the PSF and/or Gamma functions to consider when computing partially coherent images.

Diffraction Limited   If checked, aberrations are ignored. Apertures are still considered.

Wavelength   The wavelength number to be used in the calculation.

Field   The field number indicates for which field position the optical transfer function is computed.

File   The name of the IMA, BIM, or ZBF image file. The IMA and BIM files must reside in the

<images> folder, while ZBF files must reside in the <pop> (see "Folders").

See the discussion section for a full description of the supported file formats.

Edit IMA File   Pressing this button will invoke the Windows Notepad editor which allows modification of the currently selected IMA file. BIM and ZBF files may not be edited directly.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String". The contours are defined in units of relative intensity.

Use Polarization   If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Resample Image   If checked, the final image will be resampled to simulate the image recorded by a detector with finite resolution. The controls below are used to define the resampling. This feature is only available for coherent, incoherent, raw, and partially coherent spatial domain images.

Number X/Y   The number of X and Y direction pixels to use for resampling the image.

Delta X/Y   The width/height of the pixels used to resample the image. In focal mode, the Delta values are measured in Lens Units. In afocal mode, the Delta values are measured in Afocal Mode Units.

Decenter X/Y   The X/Y decenter of the resampled detector relative to the final image. The units are resampled delta pixels. For example, a Decenter X of +0.5 will decenter the resampling detector one half the Delta X value to the right relative to the final image.

Output File   If the input file is a ZBF file, and either a fully coherent or incoherent image is selected for the Data Type, then the resulting complex amplitude may be saved in a ZBF file of the name specified in this control. If the Output File name is left blank, no file is saved. No path name should be provided, as the file will be saved in the <pop> folder (see "Folders"). To save ZBF files, the zero pad must be set to yield a total number of pixels that is a power of 2, as this is required for the ZBF file format.

Discussion

This feature can compute complex diffraction image properties from extended sources. The method involved in the computation is based upon Fourier Optics, which is described in clear and insightful detail in "Introduction to Fourier Optics" by Joseph Goodman, McGraw-Hill 1968. A more detailed treatment of coherent imaging theory may be found in "Linear Systems, Fourier Transforms, and Optics" by Jack Gaskill, John Wiley 1978. See those references for more information on coherent vs. incoherent imaging and other Fourier optics theory. There are several important assumptions in the method that must be understood by the user before this feature may be used to draw important conclusions.

The file size parameter determines the full width of the input file in the image space of the optical system. For details on the IMA format see "The IMA format". For details on the BIM file format see "The BIM format". For details on the ZBF file format see "Zemax Beam File (ZBF) binary format" and "Zemax Beam File (ZBF) text format".

Note this feature is different from the geometric image analysis feature, where the file size defines the size and shape in "field" space, which may be either object or image space. For this feature, the input file defines the ideal image in image space. The data values in the IMA or BIM file are interpreted to be in units of relative irradiance, not amplitude. The ZBF file is in units of relative complex amplitude.

Comment about using the IMA file format

Although the IMA file is convenient for defining simple shapes, the resolution of files created by hand is generally too low to see sufficient detail in the diffracted image. The oversampling option remedies this problem by increasing the resolution; the number of pixels is increased, and the data from each pixel is replicated as required to yield the same shape at higher resolution. With an oversampling of 2X; the letter "F" file IMA image becomes a 14 x 14 grid:

00111111111100

00111111111100

00110000000000

00110000000000

00110000000000

00110000000000

00111111110000

00111111110000

00110000000000

00110000000000

00110000000000

00110000000000

00110000000000

00110000000000

Note the shape is the same, there is just twice the sampling in each direction. The width of the image is unaffected; the pixels are half the width and there are twice as many in each direction now.

Because the effects of diffraction tend to blur and extend the ideal image, it is desirable to increase the size of the displayed area beyond the limits of the IMA file. This can be done with the zero padding option. This option increases the IMA file size by adding zero intensity values around the defined IMA file pixels. The letter "F" file would look like this with a zero padding of 2X:

0000000000000

0000000000000

0000000000000

0000111110000

0000100000000

0000100000000

0000111100000

0000100000000

0000100000000

0000100000000

0000000000000

0000000000000

0000000000000

The size and shape of the image is the same, but additional area has been defined that some energy may now diffract into. Note the width of the new image has increased by the zero padding factor, but the width of the "F" part of the image is the same. If the original file has an odd number of pixels (as this sample letter F does) then the number of pixels after zero padding will remain odd to preserve the notion of a "center" pixel. The number of total pixels is thus 2*n-1 if n is odd, or 2*n if n is even, where n is the number of original pixels. Zero padding is not available when using the binary IMA file format.

The oversampling and zero padding may be used together, however the array sizes quickly become quite large. For an oversampling of 8X, a zero padding of 4X, and an original pixel size of 12 x 12, the resulting array becomes 384 x 384, which Zemax internally further zero pads up to 512 x 512 for transform purposes.

Comment about using the ZBF file format

When using the ZBF file format for this feature, note that only the complex amplitude of the unpolarized beam is considered. None of the other data in the ZBF file is relevant, including the pilot beam data, wavelength, and pixel size. If the ZBF file has an unequal number of pixels in each direction, the file size is the larger of the two dimensions, and zero-padding is added to make the file square. The size of the pixels as defined in the ZBF file are not used. The ZBF file format is only used here for convenience in specifying complex amplitude, and most of the features specific to physical optics analysis are not used.

When performing an incoherent image analysis using a ZBF file as input, the intensity of the data is computed by summing the squares of the real and imaginary values at each point. The phase information is thus lost, as is correct for incoherent image formation. The resulting image is a real valued intensity image. When performing coherent image analysis, the complex amplitude as defined in the ZBF file is used.

The optical transfer function algorithm

Once the input image is defined, the image is transformed into frequency space, multiplied by the OTF, and transformed back into position space. The primary assumption made by this implementation of the Fourier method

is that the OTF does not change over the extent of the image region (to avoid this assumption see "Image Simulation"). This means the field of view defined by the size of the image is small enough so that the OTF is the same over all points in the image. The user must take care to be sure the image region defined is small with respect to the rate of change of the field aberrations. Zemax computes the OTF for the field point selected, and assumes this OTF is valid over the entire region covered by the image.

Because of this assumption, distortion will not be visible in the predicted image, since only the variation in OTF over the field will introduce distortion. To see the effects of distortion or other "large field" effects, see "Image Simulation".

Note that this feature is good at computing detailed image data for small images under partially coherent light, while the Image Simulation feature is good at computing image data for larger scale images or incoherent light.

Comment about the coherent optical transfer function

The other assumption this calculation makes is in the method used to compute the coherent optical transfer function. The coherent optical transfer function is assumed to be the complex pupil function:

$$H(f_x, f_y) = P(\lambda d_i f_x, \lambda d_i f_y) e^{iKW(\lambda d_i f_x, \lambda d_i f_y)}$$

where H is the complex OTF, d_i is the pupil to image distance, f_x and f_y are the spatial frequencies, P is the pupil function (which determines the relative transmission over the pupil, and is zero outside the pupil), and W is the wavefront aberration function. This approximation is valid for a wide range of optical systems. For more information on the development of coherent imaging theory, see the references by Gaskill or Goodman.

This feature also accounts for any vignetting at the image surface due to surface apertures (apertures at other surfaces are accounted for via the effects of those apertures on the optical transfer function). After the diffraction image is computed, any energy that lies outside of the aperture defined on the image surface is eliminated. The fraction of energy within the unvignetted portion of the image surface aperture is reported on the text version of the analysis, if the image surface has a defined aperture.

Computing partially coherent images

Note: This feature is only available in the Professional and Premium editions of Zemax.

The prior discussion applies to fully incoherent or fully coherent illumination. Partially coherent imaging is more complex. The method Zemax uses to model partially coherent illumination is described in B. Saleh and M. Rabbani, "Simulation of partially coherent imagery in the space and frequency domains and by modal expansion", Applied Optics Vol. 21, No. 15, p2770 (1982).

The diffraction image for the partially coherent case can be described as:

$$I_i(\vec{r}) = \iint \sqrt{I_0(\vec{r}_1)}\sqrt{I_0(\vec{r}_2)}\, \gamma(\vec{r}_1 - \vec{r}_2) h * (\vec{r} - \vec{r}_1) h(\vec{r} - \vec{r}_2) d\vec{r}_1 d\vec{r}_2$$

where $I_i(\vec{r})$ is the irradiance of the image $I_0(\vec{r})$ is the irradiance of the unaberrated, undiffracted image of the object in image space, $\gamma(\vec{r})$ is the complex degree of coherence in image space, $h(\vec{r})$ is the complex point spread function in image space, and the symbol * indicates the complex conjugate. The position vector $\vec{r}$ has components x and y in image space. The function $h(\vec{r})$ is assumed to be invariant over the entire region of the image computed, which is the same assumption made in the fully coherent and incoherent cases.

The key difference between the fully coherent or fully incoherent illumination and partially coherent illumination is contained with the function $(\vec{r})$, which defines the degree of coherence between any two points in the object illumination. The function $\gamma(\vec{r})$ is symmetric, with a magnitude between 0 and 1 for all $\vec{r}$. For fully incoherent imaging $\gamma(\vec{r}) = \delta(\vec{r})$, where $\delta(\vec{r})$ is the Dirac delta function. For fully coherent imaging, $\delta(\vec{r}) = 1$ for all $\vec{r}$. In either of these two extremes, the general imaging equation above may be substantially simplified. For the partially coherent case, no easy simplification is possible, and for this reason partially coherent computations are generally much slower than fully coherent or incoherent cases.

To speed up the partially coherent image computation, two distinct algorithms may be employed to solve the imaging integral above. The two algorithms are user selectable. If the Mostly Incoherent method is selected, the algorithm assumes the function $\gamma(\vec{r})$ has a spatial extent that is small compared to the width of the image being computed. If the Mostly Coherent method is selected, the algorithm assumes the spatial extent of the function $\gamma(\vec{r})$ is large, or at least comparable to the width of the image. The exact computation speed depends upon the size and structure of both $h(\vec{r})$ and $\gamma(\vec{r})$, as well as the function $I_0(\vec{r})$. A good rule of thumb is that if $\gamma(\vec{r})$ is wider than about 20% of the image, the Mostly Coherent method is faster. Some experimenting may be required to determine the faster method for any specific case. Both of these algorithms are multi-threaded and will automatically use parallel execution on any number of available CPU's for maximum computation speed.

Two different functional forms for $\gamma(\vec{r})$ are currently supported. The Gaussian form is:

$$\gamma(\vec{r}) = e^{-\left(\frac{\vec{r} \cdot \vec{r}}{\alpha^2}\right)}$$

The Sinc form is:

$$\gamma(\vec{r}) = SINC\left(\frac{x}{\alpha}\right) SINC\left(\frac{y}{\alpha}\right), \text{ where } SINC(x) = \left(\frac{sin\pi x}{\pi x}\right)$$

Both the Mostly Coherent and Mostly Incoherent methods accelerate the computation by assuming the various functions have a finite width that has a significant effect on the final image. This finite width is defined by the function and by the user defined parameter called "Fraction". This is the fraction of total energy within the functions $h(\vec{r})$ and $\gamma(\vec{r})$ that is considered significant. Zemax will determine the spatial dimension of significance of these functions using the defined fraction. The closer the fraction is to unity, the slower the computations will be. Generally, values between 0.90 and 0.96 are recommended. Values larger than 0.96 may substantially increase the computation with little effect on the final image.

Two additional "Data Type" options are avaliable for use with partially coherent computations. The "Partially Coherent Test: PSF" will display the diffraction PSF (this is the intensity of the function $h(\vec{r})$) used for the partially coherent image computation. The various sampling and image width settings affect the sampling used for the PSF. Before any partially coherent image can be considered accurate, the PSF should be computed to confirm adequate sampling of the PSF. At least 10 points should be displayed over the width of the non-zero portion of the PSF for adequate sampling. The data type "Partially Coherent Test: Gamma" displays the magnitude of the function $\gamma(\vec{r})$.

Computing MTF using Cross Section X/Y

The "Show As" option supports "Cross Section X" and "Cross Section Y" options. These options compute the same complete diffraction image, but then plot only a single cross section across either the columns or rows of the image, for X and Y, respectively. The intended purpose of these cross sections is to visualize the effective MTF of the image of a bar target. The MTF can be estimated by determining the maximum and minimum relative intensity across the cross section. To reduce the effect of edges, the analysis parameters should be set to provide at least 5 well defined peaks across the cross section. The MTF is computed by looking for the minimum and maximum intensity at all points between the second and second-to-last local peaks in the intensity data. By considering only data within these two peaks, the effects of the edges is somewhat reduced. The estimated MTF is then given by the usual computation of (Imax-Imin)/(Imax+Imin). Finally, note that if a bar target is used the resulting MTF is the square-wave, not sine-wave modulation.

Extended Diffraction Image Analysis

The feature "Image Simulation" is generally superior to this feature.

The extended diffraction image analysis feature is similar to "Partially Coherent Image Analysis", except the Optical Transfer Function (OTF) may vary over the field of view of the image and the illumination must be fully coherent or incoherent. This feature uses IMA/BIM files to describe the object to be imaged. See "The IMA format" and "The BIM format"
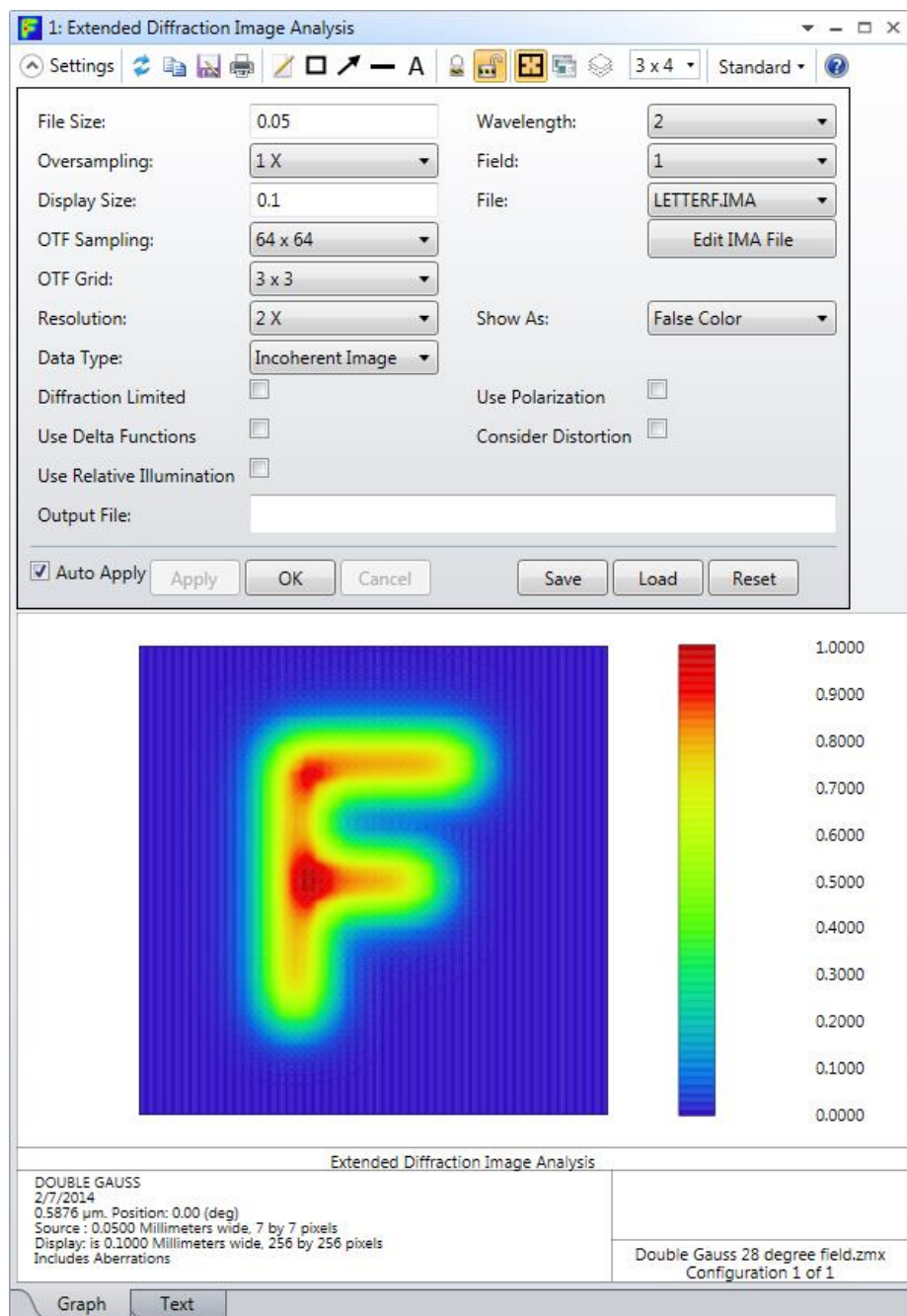
File Size The full width in lens units of the region defined by the IMA file. Note IMA files are always square. This control does not set the size of the resulting image; that is defined by the Display Size.

Oversampling   Sets the factor by which the IMA file pixels are oversampled. This increases the effective resolution of the IMA file without the need to define a new IMA file. If the original IMA file has an odd number of pixels, oversampling will make the number of pixels even because the oversampling values are all even.

Display Size   This defines the width of the displayed image in lens units.

OTF Sampling   The grid size of the sampling in the pupil; larger grids yield a more accurate representation of the system OTF. Larger grids will also yield a larger maximum display size (see the discussion below).

OTF Grid   The grid size of the OTF computation. A denser OTF grid yields a more accurate computation of the variation of OTF over the field of view of the image at the expense of more memory usage and slower computation.

Resolution   This multiplier yields more points in the final image while keeping the display size fixed, at the expense of more memory usage.

Show As   Choose surface plot, contour map, grey scale, or false color map as the display option.

Data Type   Choose either the incoherent image or coherent image. See "Comment about the coherent optical transfer function" for information about the coherent transfer function and the limitations of this computation.

Diffraction Limited   If checked, aberrations are ignored. Apertures are still considered.

Use Delta Functions   If checked, each pixel in the IMA file will be assumed to represent a delta function. This is useful for checking the imaging of point sources like stars. If unchecked, the whole pixel is assumed to be a luminous square area.

If Use Delta Functions is checked, it is very difficult to visually see the effects of aliasing in the computation. For this reason, it is highly recommended that this option be unchecked while setting up the file size, display size, and sampling parameters. Once the proper values for these settings are determined, then Use Delta Functions may be checked if desired.

File   The name of the image file. This file must reside in the <images> folder (see "Folders"). See "The IMA format" and "The BIM format" for a description of file formats.

Edit IMA File   Pressing this button will invoke the Windows Notepad editor which allows modification of the currently selected IMA file. BIM files cannot be edited this way.

Wavelength   The wavelength number to be used in the calculation.

Field   The field number indicates for which field position the image will be centered on.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String". The contours are defined in units of relative intensity.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Consider Distortion   If checked, the real vs. paraxial ray distortion will be considered in forming the image appearance.

Output File   If a name is provided, the complex amplitude of the resulting image will be saved in a ZBF file with the specified name. The file will be saved in the <pop> folder (see "Folders"). See "Zemax Beam File (ZBF) binary format".

Use Relative Illumination   If checked, then the computation described in "Relative Illumination is used to weight the OTF over the field of view to accurately account for the effects of exit pupil radiance and solid angle.

Discussion

This feature can compute complex diffraction image properties from extended sources while accounting for the variation in the optical transfer function (OTF) over the field of view. Most of the discussion in "Partially Coherent Image Analysis" also applies to this feature. The differences are described below.

Diffraction image formation can be thought of as a filtering or as a convolution process. Suppose the ideal, unaberrated, undiffracted image is described by a function "A" which describes image amplitude as a function of spatial coordinates

in the image space of an optical system. Convolving this function with the system PSF (see "FFT PSF") here denoted by "P" yields the final image "I":

$$I(x, y) = A(x, y) \otimes P(x, y)$$

where the notation $A \otimes P$ denotes the convolution of A and P. Taking the Fourier transform of this equation yields the spatial filtering perspective of the image forming process:

$$i(f_x, f_y) = a(f_x, f_y) \times o(f_x, f_y)$$

where i, a, and o are the transforms of I, A, and P into the spatial frequency domain. The function o is called the optical transfer function (OTF); which acts as a filter scaling the amplitude and phase of the spatial frequency components of the image.

The Extended Diffraction Image Analysis eliminates one major assumption of the Partially Coherent Image Analysis feature: that the OTF is constant over the field of view represented by the function A. This is accomplished by considering the source IMA file one pixel at a time, and computing the Fourier transform of the one pixel. The one pixel transform is multiplied by the OTF corresponding to that pixel. The sum over all pixels is then computed in spatial frequency space, and finally the sum of the filtered pixel transforms is Fourier transformed back to form the final image.

Computing the OTF for every pixel in the IMA file is slow and not practically required, as the OTF over any reasonably small field of view does not change rapidly. As an alternative, this feature computes a grid of OTF which spans over the field of view, and then interpolation is used to compute the effective OTF at any single pixel. The greater the size of the OTF grid, the more accurate the results will be, at the expense of longer computation times and more memory usage.

This feature can require large amounts of RAM. For an OTF grid size of n x n, Zemax will actually store n*n+1 OTF's: the n*n OTF's on the grid plus one to use as a running sum of the image frequency components. The OTF itself is a complex value array of double precision points. Zemax automatically zero pads the OTF grid to achieve accurate sampling. If the OTF sampling is 64 x 64, a single OTF will be 128 x 128, using up 256k of RAM (128 x

128 x 2 x 8 bytes). For an OTF grid of 5 x 5, the total comes to 26*256k = 6.5 Mb. For 128 x 128 sampling and a 9 x 9 grid, the RAM required is 82 Mb.

The file size parameter determines how big each pixel in the IMA file is in the image space of the optical system. Note this is different from the geometric image analysis feature, where the IMA file defines the size and shape in "field" space. For this analysis, the IMA file defines the ideal image shape in image space. If the file size is 0.1 mm, then each pixel is 14.286 micrometers wide for this 7 x 7 pixel image.

Relative Illumination

The Relative Illumination analysis computes the relative illumination as a function of radial field coordinate for a uniform Lambertian scene. This feature also computes the Effective F/#.
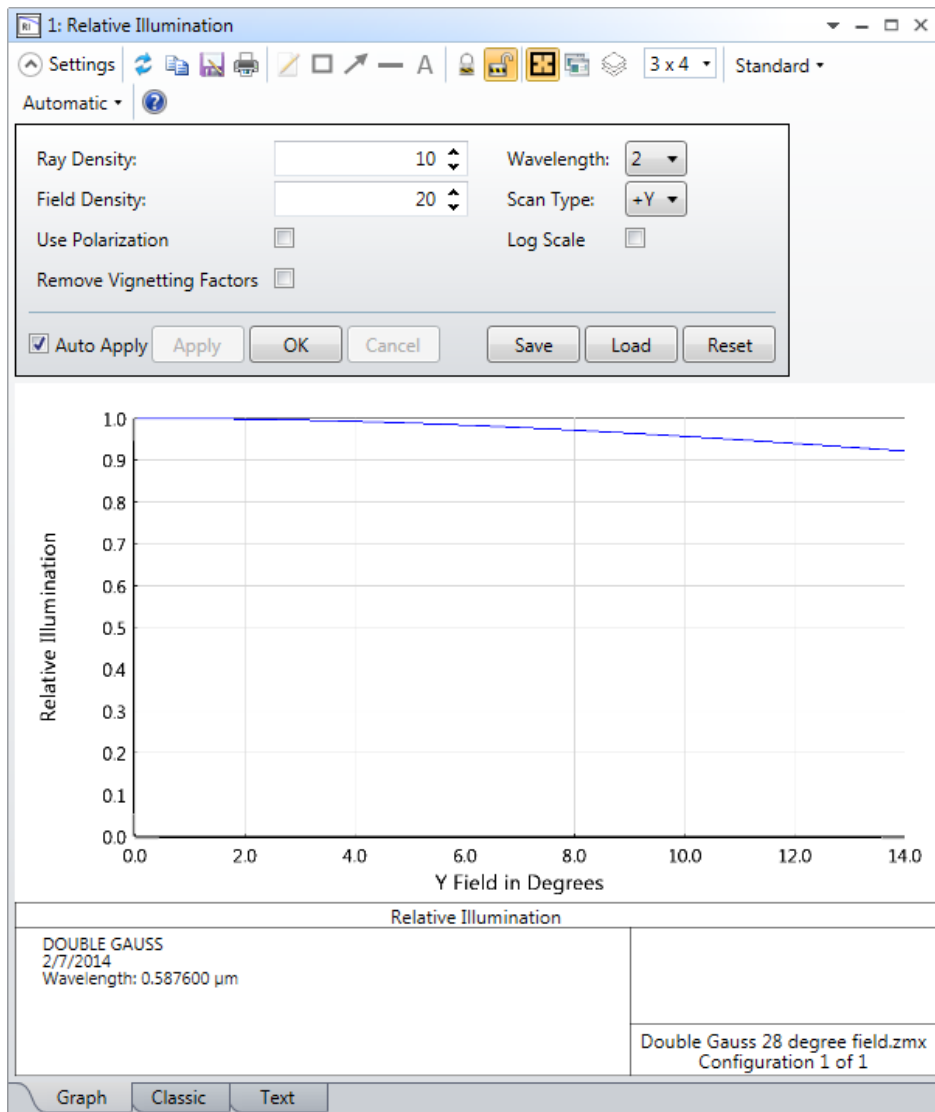
Ray Density The number of rays on one side of an array of rays used to integrate the illumination of the exit pupil. A value of 10 will trace about 10 x 10 x pi / 4 or 78 rays. Higher ray densities yield more accurate results at the expense of longer computation times.

Field Density The number of points along the radial field coordinate to compute the relative illumination for. Larger field densities yield smoother curves.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Remove Vignetting Factors If checked, vignetting factors are automatically removed. See "Comment about vignetting factors" in the description of "FFT MTF vs. Field".

Wavelength Selects the wavelength for computation. Relative illumination is a monochromatic entity.

Scan Type Choose +y, +x, -y, or -x field scan direction.

Log Scale If checked, a logarithmic rather than linear scale will be shown.

Discussion

This feature computes the relative illumination (RI) as a function of radial y field coordinate. RI is defined as the intensity of illumination per unit area of image surface normalized to the illumination at the point in the field that has maximum illumination (which may not be on axis). The computation considers apodization, vignetting, apertures, aberrations of both the image and pupil, variations in F/#, chromatic aberrations, image surface shape, angle of incidence, and optionally, polarization effects assuming unpolarized light. The method is based upon one described in M. Rimmer, "Relative illumination calculations", Proc. SPIE Vol. 655, p99 (1986). The published method was extended to include apodization, transmission, polarization, and non-planar image surface effects. The computation method assumes the following are all true:

1. The object scene is plane, uniform, and Lambertian.

2. The image surface is a reasonably good conjugate (that is, an image) of the object surface, so that rays coming from small patches of light on the object surface are imaged to patches of light on the image surface. Aberrations are fine, but the rays should be reasonably localized on the image surface.

3. The exit pupil is not too close to the image surface. This condition will be satisfied if the F/# is larger than about 0.1 and the ray aberrations are small compared to the exit pupil distance.

4. The cosine space aberrations are not so severe as to form caustics in angle space. A caustic in angle space means that rays in different parts of the entrance pupil have the same angle in image space. To check this, use the "Direction Cosines" option on the spot diagram feature (See the "Standard Spot Diagram" analysis).

The relative illumination is computed by integration of the effective area of the exit pupil as seen from the image point(s). The integration is carried out in direction cosine space using a uniform grid in image cosine space.

Note that the RI computation will not in general yield a cosine-fourth curve, because the so-called cosine-fourth "law" is actually an approximation based upon a thin, slow, aberration free lens with the stop at the lens. For more general lenses, including telecentric, aberrated, or vignetted lenses, the RI can be computed using an integration of the projected solid angle or effective area of the exit pupil as seen from the image location, and this computation will not yield a simple cosine-fourth curve. If a system violates the assumptions of the computation an error message will be displayed and the RI will not be computed.

Effective F/#

The text listing of the relative illumination data also includes data for the Effective F/#. The Effective F/# is the F/# required for a perfect optical system with 100% transmission and a circular exit pupil to have the same image

illumination as the system being evaluated. The Effective F/# is computed by:

$$E = \sqrt{\frac{\pi}{4A}},$$

where A is the area of the projected solid angle of the pupil in cosine space weighted for system transmission. The Effective F/# is a useful metric for comparing the brightness of the image formed by different optical systems, because it accounts for RI and is independent of the aperture shape. For more information see "F-Number and the radiometry of image forming optical systems with non-circular aperture stops," R. Siew, Proc. of SPIE Vol. 5867 (2005).

IMA and BIM file viewer

This feature displays IMA/BIM files. See "Geometric Image Analysis" for a discussion of IMA/BIM
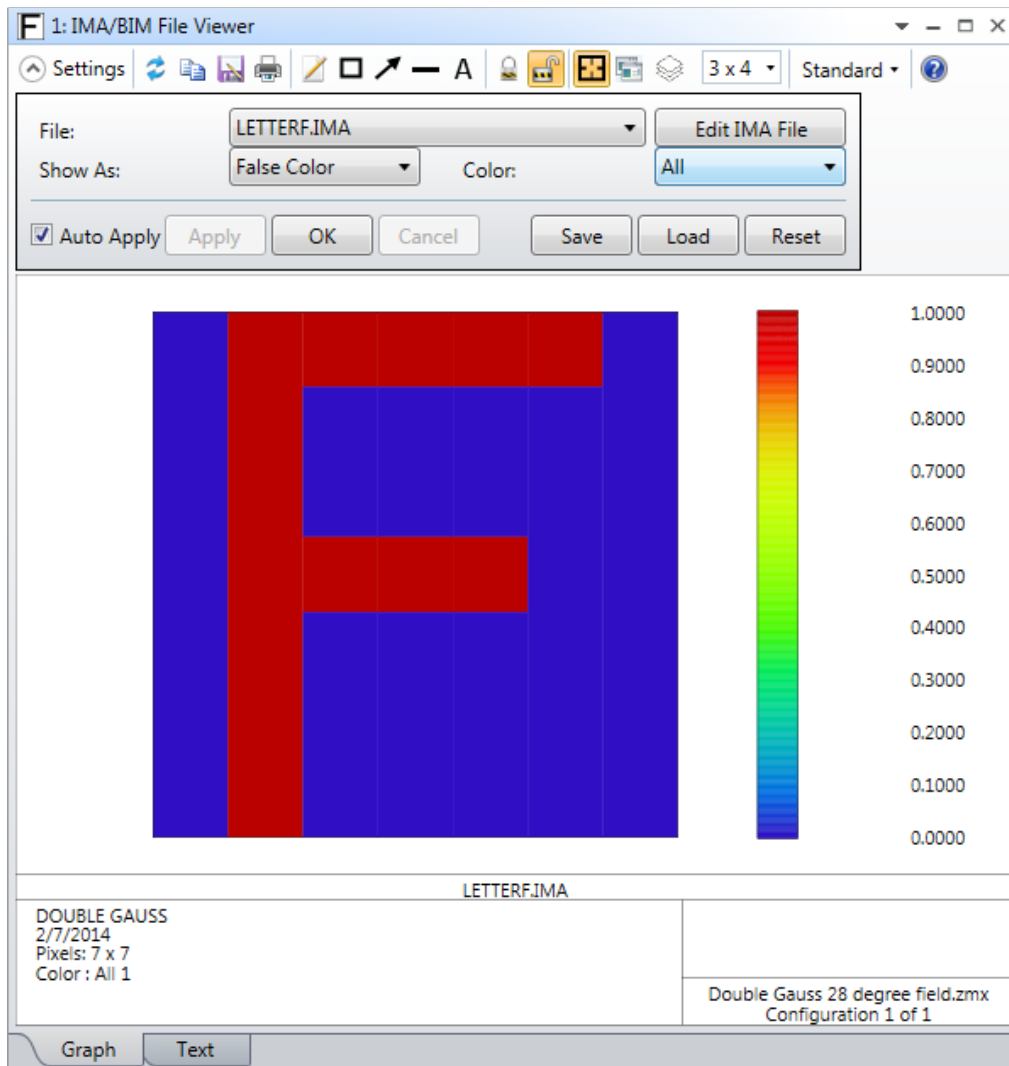
files.

File The name of the .IMA or .BIM image file. This file must reside in the <images> folder (see "Folders"). See the discussion section for a full description of the IMA and BIM file formats.

Show Choose surface plot, contour map, grey scale, or false color map as the display option.

Color If the selected file is a binary IMA, this control will select the color channel to display.

Edit IMA File Pressing this button will invoke the Windows Notepad editor which allows modification of the currently selected IMA file. This button is disabled if the file type is BIM.

Discussion All input values are normalized to a relative intensity of 1.0 for display purposes. See "Geometric Image Analysis" for a discussion of IMA/BIM files.

Bitmap File Viewer

This feature displays JPG, BMP, and PNG format image files without any processing.
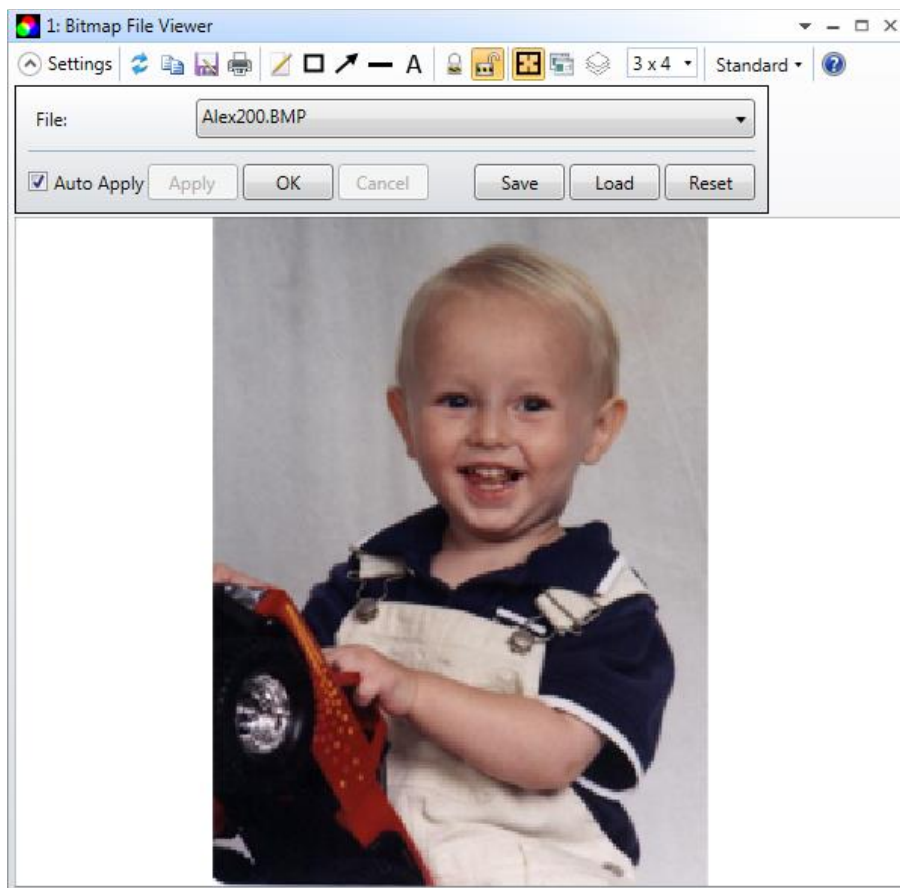
File   The name of the BMP, JPG or PNG image file. This file must reside in the <images> folder (see "Folders").

Laser and Fibers Group
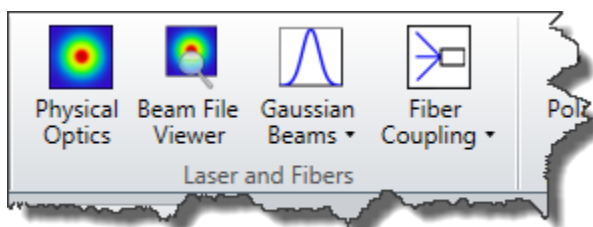
Physical Optics Propagation

This feature propagates an arbitrary coherent optical beam through the optical system. For a complete description of this feature see "PHYSICAL OPTICS PROPAGATION". The Physical Optics Propagation feature results are generally not accurate for systems including non-sequential component groups; see "Propagating through non-sequential surfaces" for more information. For the most accurate results, replace non-sequential component groups with sequential equivalents.
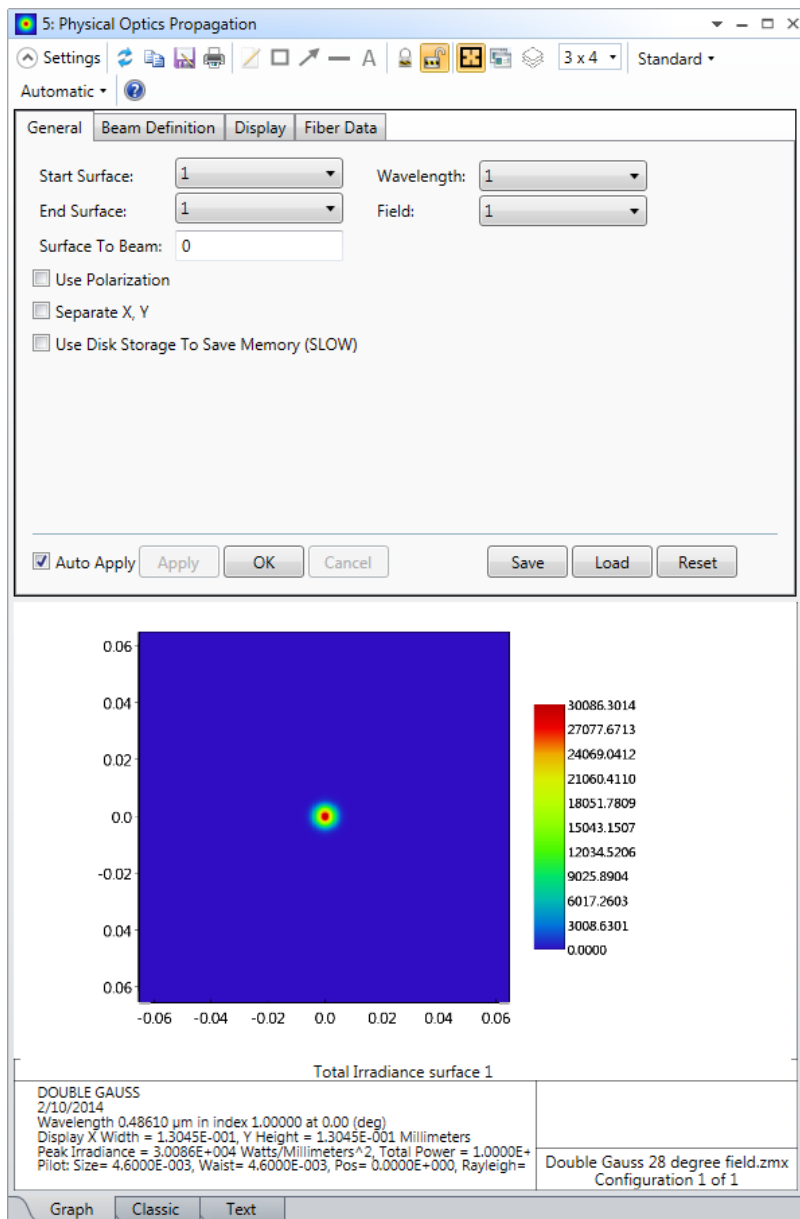
General Tab

Start Surface   The starting surface for the initial beam. The beam will begin in the optical space just prior to the starting surface.

The entrance pupil may be used as the starting surface. If the system is telecentric in object space, then surface 1 will be used as the starting surface even if the entrance pupil is selected.

End Surface   The surface at which to terminate the propagation. The beam will stop just after entering the optical space of the end surface.

Wavelength   The wavelength number to use for the beam.

Field   The field number of the chief ray used to align the initial beam.

Surf to Beam   The distance from the starting surface to the starting beam position in lens units. This value is negative if the beam starts to the left of the surface.

Use Polarization If checked, 2 beam arrays will be propagated, one each for the X and Y polarizations of the beam.

Separate X, Y If checked, the X and Y directions are propagated independently. This allows much more efficient sampling if the beam is astigmatic or the optics are cylindrical or toroidal.

Use Disk Storage To Save Memory See "Memory requirements" for a discussion.
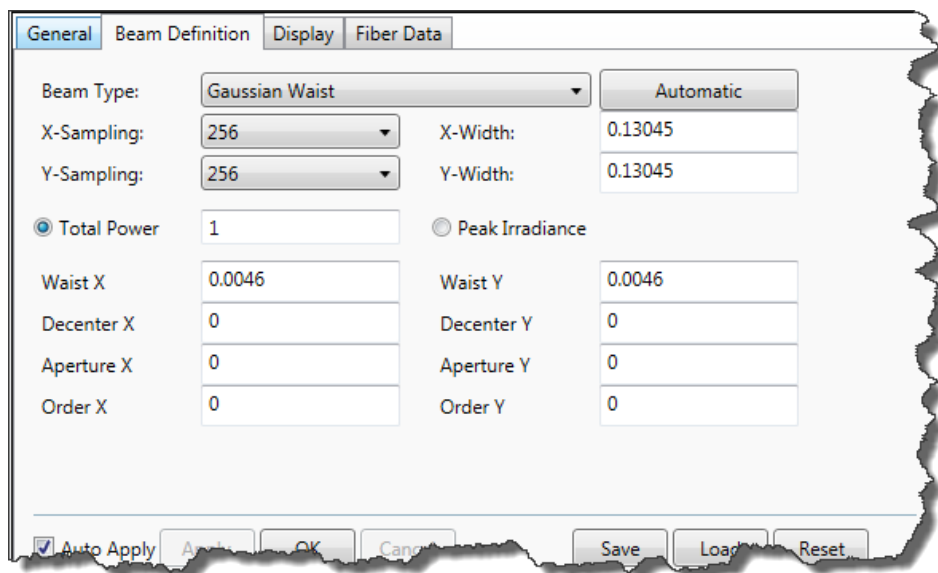
Beam Definition Tab

X- Y- Sampling The number of points used to sample the beam.

X- Y- Width The initial width in lens units of the region represented by the array. See Auto below.

Auto Pressing this button will calculate the optimum X- and Y- Width values to maintain the approximately the same number of pixels across the beam both within and outside the Rayleigh range. See "Comments about point spacing and sampling".

Type See the discussion and "Defining the initial beam".

File See the discussion and "Defining the initial beam".

Peak Irradiance The initial beam peak irradiance in power per area. This is an alternative to Total Power.

Total Power The initial beam total power. This is an alternative to Peak Irradiance.

Read power from file If the beam type is File, pressing this button will read the current power and peak irradiance values from the selected file.

Parameters Different initial beam types require various parameters to define the distribution. These parameter names and values will change depending upon the beam type setting. For details, see "Defining the initial beam".
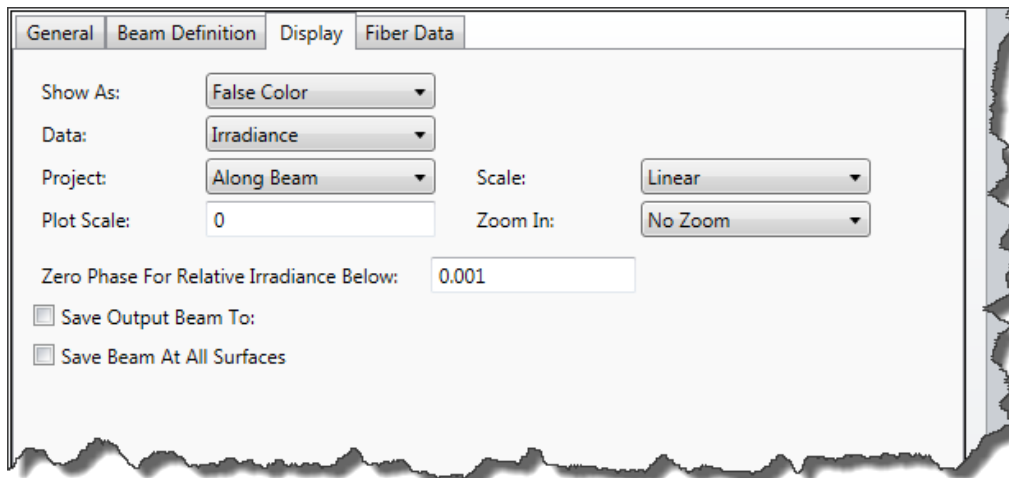
Display Tab

Show As   Choose surface, contour, false color, grey scale, cross sections, encircled, ensquared, or enslitted displays. Note that the encircled energy is a function of radius; and the ensquared and enslitted widths are the half widths.

Row/Column   If cross section is chosen for Show As, this control selects the row or column to view.

Data   Choose irradiance (beam power per area), phase (of the beam), transfer function intensity, or transfer function phase. The Ex and Ey refer to the X and Y polarization components of the beam. The transfer function values are the intensity and phase of transmission for the final surface only; these are primarily used for debugging and analysis of the POP results.

For text POP windows only, a propagation report is also available. The propagation report lists various data about the propagation algorithm parameters used, the orientation matrix for the pop results, and may contain warnings that indicate if a possible inaccuracy is detected in the results.

Scale   Choose linear or logarithmic scaling of the data. Logarithmic scaling only applies to irradiance data on graphical (not text) displays.

Project The beam may be viewed from any one of these perspectives: along the beam (this is along the chief ray used by the pilot beam), along the surface normal, or along the local Z axis of the surface. Note ZBF beam file data is always "along the beam". See the comments about beam projection below.

Save Output Beam To   If checked and a file name is provided, the complex amplitude of the beam at the end surface will be written to a file. This file can be read back in as a starting beam using the Beam Type control. Zemax will add the extension "ZBF" to the provided file name. ZBF files are stored in the <pop> folder (see "Folders").

Save Beam At All Surfaces   If checked, the beam file will be saved at every surface from the start surface to the end surface that uses the physical optics propagator. The "Save Output Beam To" option must also be checked and a file name provided. The individual surface beam files will be named according to the provided file name, with a suffix indicating the surface number. For example, if the output file name is "MyBeamData.ZBF", then the beam file name for surface 12 will be "MyBeamData_12.ZBF". ZBF files are stored in the <pop> folder (see "Folders").

Zero Phase For Relative Irradiance Below   Phase values are meaningless if the irradiance is extremely low. Computing the phase angle of data points with nearly zero irradiance (relative to the peak irradiance in the beam) will result in noisy plots and text listing of meaningless data. This value sets the lower limit on the relative irradiance of data points for which the phase is computed. Data points with relative irradiance lower than this threshold will have a phase value of zero.

Plot Scale   If zero, the vertical scale on false color, grey scale, and cross section plots will be set by the maximum data value, or for log plots, the next largest power of 10. Otherwise, the maximum vertical scale is set by the normalization value. If a log plot is generated, the maximum scale is set by the next largest power of 10 greater than the plot scale value.

Phase plots always scale from -pi to pi.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String" . The contours are defined in the same units as the data being displayed.

Zoom In Magnifies the displayed area of the beam. This is useful for eliminating some of the guard band from the displayed beam data.
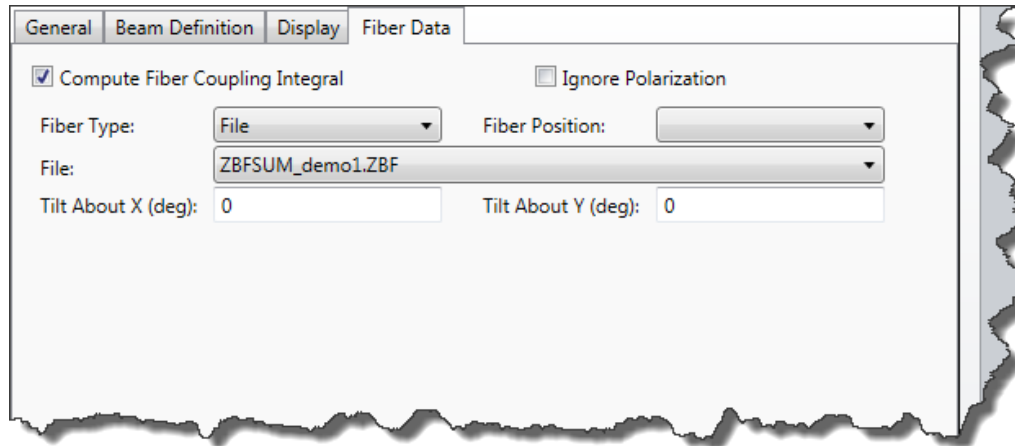
Fiber Data Tab

Compute Fiber Coupling Integral If checked, the fiber data on this tab will be used and the fiber coupling computed; otherwise, no fiber coupling computations will be performed. See "Computing Fiber Coupling".

Ignore Polarization If checked, the fiber mode will be unpolarized even if the beam itself is polarized. If unchecked and the beam is polarized, then the fiber coupling overlap integral is computed for the two orthogonal polarizations independently and the results combined to yield the overall coupling. See "Defining the fiber mode".

Tilt About X/Y (deg) The tilt about each axis in degrees of the fiber mode with respect to the beam. The phase of the fiber mode will be modified with a linear tilt proportional to the tilt angles.

Fiber Type Selects the mode for the fiber. See "Computing Fiber Coupling".

File Selects the name of the DLL or data file describing the fiber mode. See "Computing Fiber Coupling".

Parameters Different fiber mode types require various parameters to define the mode. These parameter names and values will change depending upon the fiber type setting.

Fiber Position The center of the receiving fiber may be referenced to the surface vertex or to the chief ray. The incoming POP beam is referenced to the chief ray position and angle.

Discussion

For a full discussion of physical optics propagation, see "PHYSICAL OPTICS PROPAGATION".

Comments about beam projection

The analysis computes the beam irradiance or phase on a plane tangent to the chief ray at the point where the chief ray intercepts the surface. The orientation matrix of the plane may be determined from the propagation report. The irradiance or phase data is shown after the beam refracts into, or reflects from, the end surface. The chief ray representing the center of the beam will generally intercept the surface at some angle of incidence other than zero, and the surface normal will generally be at some angle to the local Z axis. Using the "project" option on the "display" tab, the beam data may be

viewed along the beam, along the surface normal, or along the local Z axis. The latter two options are implemented by elongating the beam by the cosine of the angle between the beam and the desired projection.

The beam itself is always represented by the data array in a coordinate system normal to the chief ray. For this reason, the beam projected onto a surface by this feature may be elongated compared to the same beam stored as a file and viewed with the Beam File Viewer, see "Beam File Viewer".

Beams in Zemax are always centered on the chief ray for the selected field and wavelength. Therefore, the data in the beam file is positioned relative to the chief ray. The center point in the beam file is at the coordinate (nx/2+1, ny/2+1).

The analysis window may include some of the following data:

- Display X Width/ Y Height: The width and height of the beam array in lens units at the end surface.

- Peak Irradiance: The maximum irradiance in power per area. To set the power and area units see "Units"

- Total Power: The total power in the beam. To set the power units see "Units".

- Pilot Beam Data: The Pilot Beam data includes the radial beam size, beam waist, position, and Rayleigh range. For detailed information on the Pilot Beam see "The pilot beam".

- Fiber Coupling: If "Compute Fiber Coupling Integral" is checked, then the fiber coupling is computed and displayed. The system efficiency is the fraction of beam power that transmits from the starting surface to the surface at which the fiber coupling is computed. The receiver efficiency is the fraction of the power in the beam mode that couples into the fiber mode. The total coupling efficiency is the product of these two numbers. For more information see "Computing Fiber Coupling".

Beam File Viewer

This feature is only available in the Professional and Premium editions of Zemax.

This feature allows viewing and analysis on previously stored Zemax Beam Files (ZBF). These files may be user defined or may be generated by the Physical Optics Propagation feature, see the feature "Physical Optics Propagation". See also the Chapter "PHYSICAL OPTICS PROPAGATION". For information on the ZBF file format, see "Zemax Beam File (ZBF) binary format".
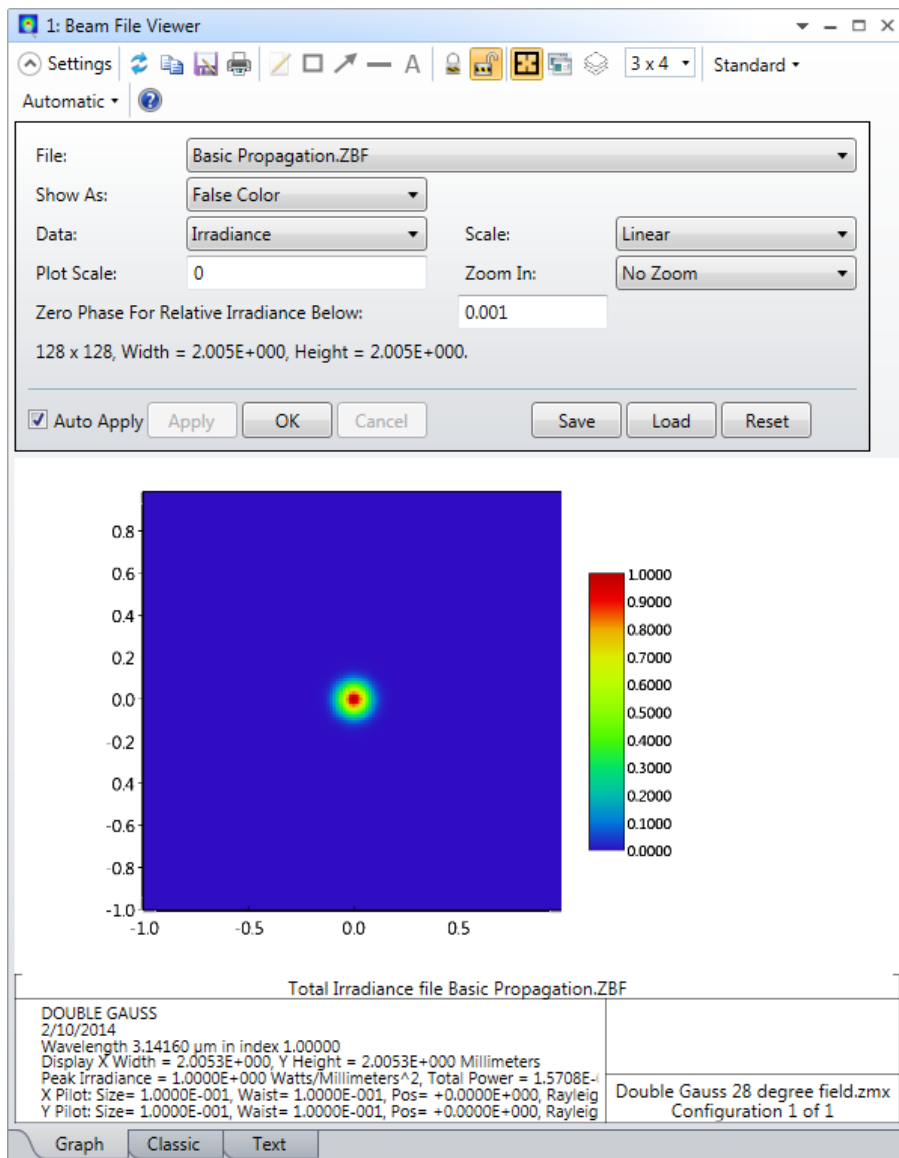
The available settings are a subset of what is available for the Physical Optics Propagation.

For rapid viewing through a range of files that differ only by the surface number, use the left and right cursor keys.

Description   The primary advantage to using this feature is that the beam may be propagated once through the optical system, using the Physical Optics Propagation feature described above, and then the beam files may be analyzed and viewed without the need to re-propagate the beam. Beam files for every surface may be generated by choosing "Save Beam At All Surfaces" on the Physical Optics Propagation feature settings.

See the "Comments about beam projection".

ZBF files are stored in the <pop> folder (see "Folders").

Gaussian Beams
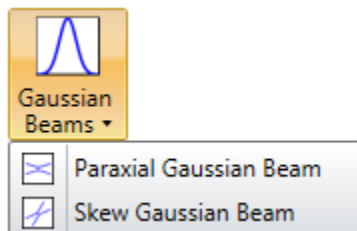
Paraxial Gaussian Beam

Computes Paraxial Gaussian Beam parameters. Paraxial Gaussian Beams are limited to axial analysis of rotationally symmetric systems. See also "Skew Gaussian Beam" and "Physical Optics Propagation"
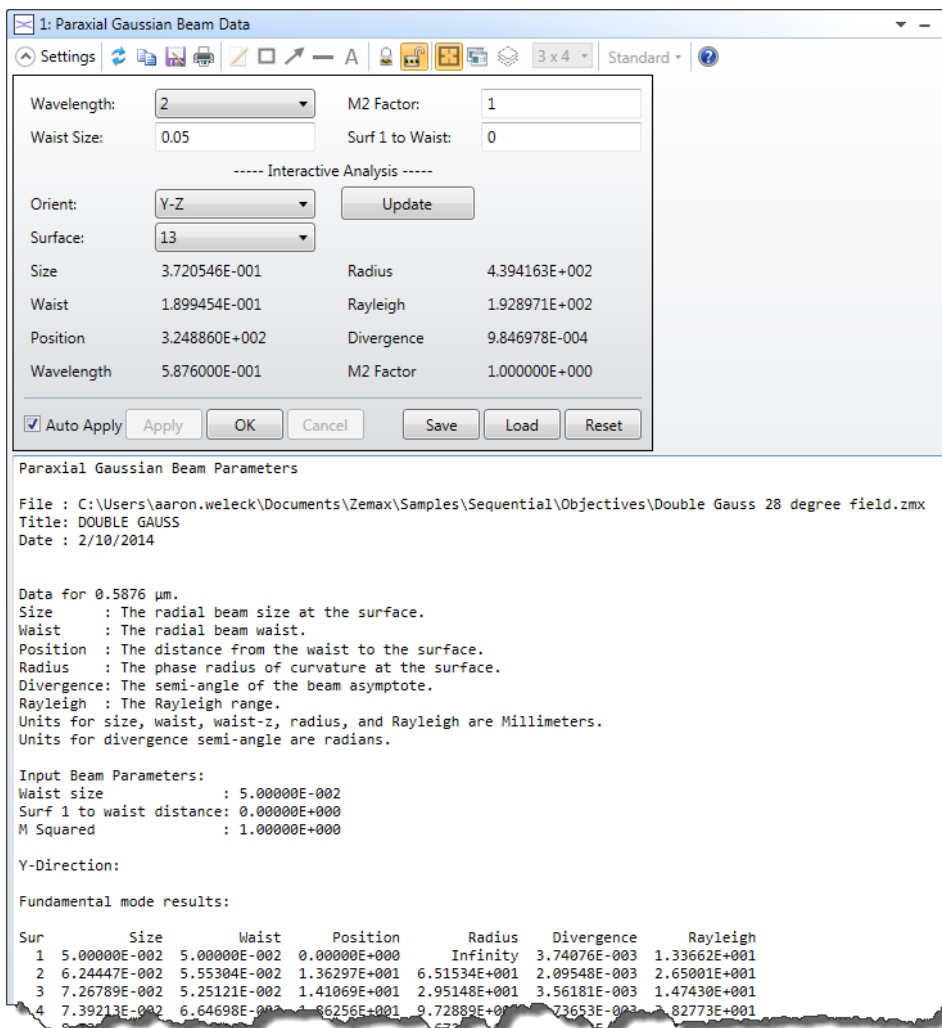
Wavelength The wavelength number to use for the calculation.

M2 Factor   The M-squared quality factor used to simulate mixed mode beams. See the Discussion.

Waist Size   The radial size of the embedded (perfect TEM00 mode) beam waist in object space in lens units.

Surf 1 to Waist The distance from surface 1 (NOT the object surface) to the beam waist location. This parameter will be negative if the waist lies to the left of surface 1.

Update See below.

Orient See below

Surface See below

Discussion

This feature computes ideal and mixed mode M-squared Gaussian beam data, such as beam size, beam divergence, and waist locations, as a given input beam propagates through the lens system. This discussion is not meant to be a complete tutorial on laser beam propagation theory. For more information on Gaussian beam propagation, see one of the following references: A. E. Siegman, Lasers, University Science Books (1986), R. Herloski, S. Marshall, and R. Antos, "Gaussian beam ray-equivalent modeling and optical design", Applied Optics Vol. 22, No. 8 p1168 (1983), M. W. Sasnett and T. F. Johnston, Jr., "Beam characterization and measurement of propagation attributes", Proc. SPIE Vol. 1414, p21 (1991), and A. E. Siegman, "New developments in laser resonators", Proc. SPIE Vol. 1224, p2 (1990).

Limitations of the analysis

Gaussian beams are idealized, "perfect" beams, and this limits the range of systems for which Gaussian beam analysis is useful. Because the calculation of Gaussian beam parameters is based upon paraxial ray data the results cannot be trusted for systems which have large aberrations, or those poorly described by paraxial optics, such as non-rotationally symmetric systems. Flat fold mirrors will not invalidate the results, but decentered or tilted components generally will. This feature ignores all apertures, and assumes the Gaussian beam propagates well within the apertures of all the lenses in the system.

For a more general beam propagation analysis feature, see "Physical Optics Propagation".

Overview of Gaussian beams

A Gaussian laser beam is described by a beam waist size, a wavelength, and a location in object space. The Gaussian beam is an idealization that can be approached but never attained in practice. However, real laser beams can be well described by an "embedded" Gaussian beam with ideal characteristics, and a quality factor, called M2, which defines the relative beam size and divergence with respect to the embedded Gaussian mode. The ideal M2 value is unity, but real lasers will always have an M2 value greater than unity.

This feature requires the definition of the initial input embedded beam properties, and the M2 value. The input embedded beam is defined by the location of the input beam waist relative to surface 1 (note this is not the object surface, but the first surface after the object) and the waist radial size at this location. Zemax then propagates

this embedded beam through the lens system, and at each surface the beam data is computed and displayed in the output window. Zemax computes the Gaussian beam parameters for both X and Y orientations.

Default beam parameters

Zemax defaults to a waist size of 0.05 lens units (no matter what the lens units are) and a surface 1 to waist distance of zero; this of course means the waist is at surface 1. The default values may be reset by clicking on the "Reset" button. After the default values are computed and displayed, any alternate beam waist size and location may be entered and that Gaussian beam will be traced instead.

Propagating the embedded beam

Once the initial beam waist and location parameters are established, Zemax traces the embedded beam through the system and computes the radial beam size, the narrowest radial waist, the surface coordinate relative to the beam waist, the phase radius of curvature of the beam, the semi divergence angle, and the Rayleigh range for every surface in the

system. Zemax calls these parameters the Size, Waist, Waist Z, Radius, Divergence, and Rayleigh, respectively, on the text listing that is generated.

The dimensions for all parameters are lens units except for the semi divergence angle, which is in units of radians. The embedded Gaussian beam parameters can be computed using the following standard formulas.

The Rayleigh range, z_r , is defined as

$$z_r = \frac{\pi \omega_0^2}{\lambda}$$

The phase radius of curvature is given by

$$R(z) = z + \frac{z_r^2}{z}$$

where z is the distance from the beam waist. The radial beam size at any z is computed from

$$\omega(z) = \omega_0 \left[1 + \left(\frac{z}{z_r} \right)^2 \right]^{1/2}$$

where ω_0 is the beam waist. The divergence angle of the beam is given by

$$\theta = \tan^{-1} \frac{\omega_0}{z_r}$$

Finally, the Gouy shift is the phase of the center of the Gaussian beam:

$$\varphi = \tan^{-1} \frac{z}{z_r}$$

The quality factor

All of the preceding results are correct for the ideal embedded Gaussian beam. For aberrated, mixed-mode beams, an extension to the fundamental Gaussian beam model has been developed by Siegman. The method uses a term called the beam quality factor, usually denoted by M2. The factor M2 can be thought of as "times diffraction limited" number, and is always greater than unity. The M2 factor determines the size of the real, aberrated Gaussian beam by scaling the size and divergence of the embedded Gaussian mode by M. Note that

the Rayleigh range and phase radius are not scaled by M, but are instead the embedded mode values as defined by Siegman. It is common practice to specify M2 for a laser beam, rather than M, although the factor M is used to scale the beam size. The M2 factor must be measured to be determined correctly. If the M2 factor is set to unity, the default value, Zemax computes the TEM00 data described above. If M2 is greater than unity, then Zemax computes both the embedded Gaussian beam parameters as well as the scaled data.

Interactive analysis

The Settings dialog box for this feature also supports an interactive mode. After defining the various input beam parameters, clicking on "Update" will immediately trace the specified Gaussian beam, and display the usual results in the dialog box. The parameters for any surface may be viewed, and the surface number selected from the drop down list. The orientation may also be selected using the provided control.

The interactive feature does not in any way modify the lens or the system data; it is a handy "calculator" for displaying paraxial Gaussian beam data.

Skew Gaussian Beam

Computes Skew Gaussian Beam parameters. Skew beams may enter an optical system at any surface from any field position, and may travel through the optical system off axis. The Skew Gaussian Beam parameters are computed using real rays and account for astigmatism but not higher order aberrations. See also "Paraxial Gaussian Beam" and "Physical Optics Propagation"
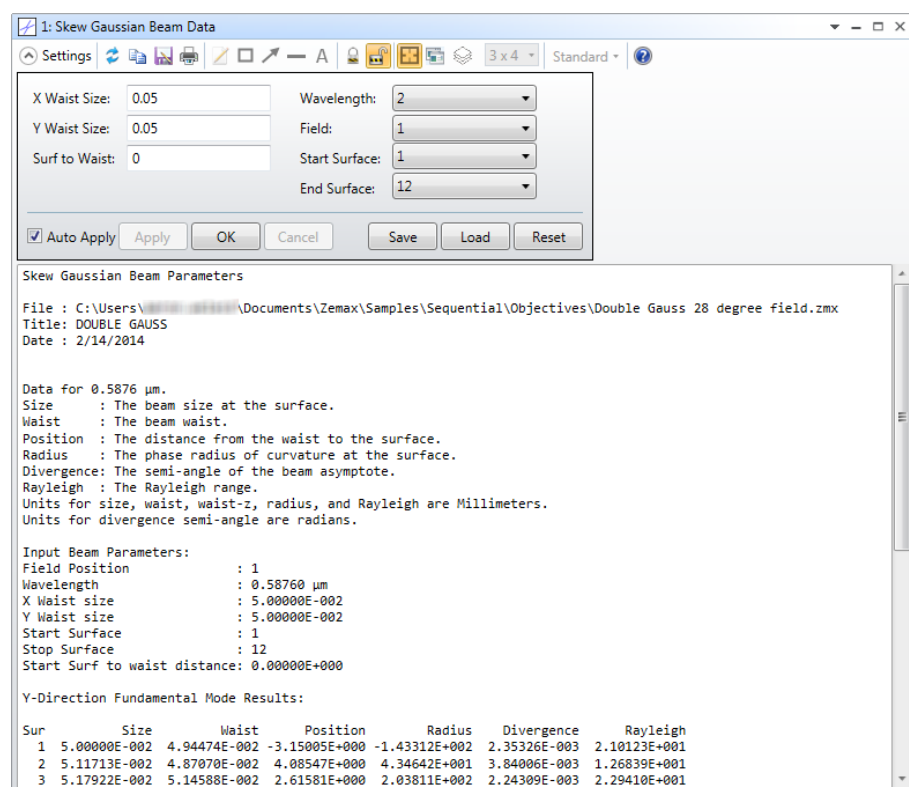
X Waist Size	The x direction radial size of the beam waist in the space prior to the "start surface" in lens units.
Y Waist Size	The y direction radial size of the beam waist in the space prior to the "start surface" in lens units.
Surf to Waist	The distance from the "start surface" to the beam waist location. This parameter will be negative if the waist lies to the left of the start surface.
Wavelength	The wavelength number to use for the calculation.
Field	The field number to use for the calculation.
Start Surface	The surface to begin the beam propagation.
End Surface	The surface to end the beam propagation.

Discussion

For important background information on Gaussian beams, see "Paraxial Gaussian Beam" .

This feature computes the ideal skew Gaussian beam data, such as beam size, beam divergence, and waist locations, as a given input beam propagates through the lens system. The beam is not limited to being on axis in a symmetric optical system, and beam at any angle may be traced anywhere in the optical system.

The beam is initially defined by a waist size in the x and y directions in the optical space prior to the starting surface. The field and wavelength settings are used to define a chief ray which is traced through the optical system. This chief ray becomes the axis for the skew Gaussian beam throughout the propagation.

The x and y directions are initially defined by the entrance pupil orientation. The +y direction relative to the chief ray is defined as a vector pointing from the chief ray coordinate at the entrance pupil to the coordinate of the px = 0, py = 1 marginal ray coordinate at the entrance pupil. The px = 1, py = 0 marginal ray defines the +x direction. For a discussion of the normalized pupil coordinates, see "Normalized pupil coordinates" .

As the chief ray propagates through the optical system, the two marginal rays are propagated as well, and are used to define the +x and +y directions in each optical space. All the data listed for x and y directions are measured in this moving coordinate system.

The Skew Gaussian Beam feature results are generally not accurate for systems including non-sequential component groups. For the most accurate results, replace non-sequential component groups with sequential equivalents.

Fiber Coupling
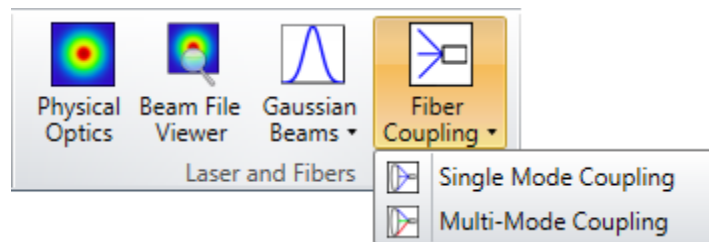
Single Mode Coupling

This feature computes the coupling efficiency for single mode fiber coupling systems. For multi-mode fiber coupling, see "Calculating efficiency of multi-mode fibers" . Also see "Computing Fiber Coupling"
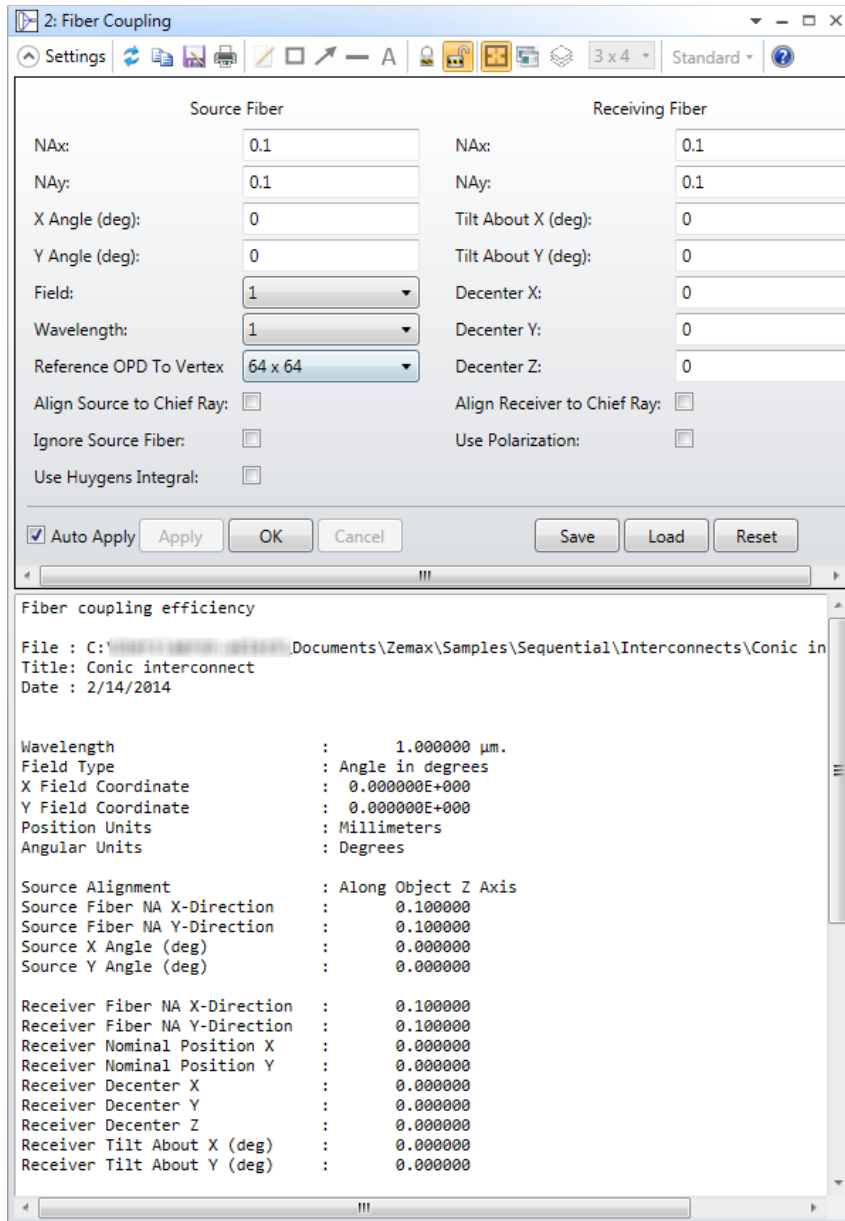
Source Fiber Settings

NA x/y Numerical Aperture of the source fiber in object space in the xz and yz planes respectively. This is n times the sin of the half angle to the 1 over e squared intensity point.

X Angle, Y Angle The angular rotation of the source fiber in object space, in degrees, measured from the nominal orientation of the source fiber. The X angle is the angle measured in the XZ plane; the Y angle is measured in the YZ plane.

Field The field number to use for the calculation.

Wavelength The wavelength number to use for the calculation.

Sampling The grid size to use for the numerical integration.

Align Source to Chief Ray If unchecked, the fiber is oriented along the object z axis, centered on the field point. If checked, the center of the fiber points along the chief ray for that field point.

Ignore Source Fiber   If checked, then the pupil is illuminated using the specified Apodization Type & Value in the Aperture Settings of the System Explorer. All computations are then referenced to the total energy incident upon the entrance pupil.

Receiving Fiber Settings

NA x/y Numerical Aperture of the receiving fiber in image space in the xz and yz planes respectively. This is n times the sin of the half angle to the 1 over e squared intensity point.

Tilt About X, Y The angular rotation of the receiving fiber in image space, in degrees, measured from the nominal orientation of the receiving fiber. If both X and Y rotations are specified, the rotation is first around X, then around Y. Note the rotations are about the specified axes; so Tilt About X rotates the fiber toward or away from the Y axis. Rotations are done after the XYZ decenters below.

Decenter XYZ   The linear position shift in the x/y/z direction in lens units of the receiving fiber, measured from the nominal position of the receiving fiber. Shifts are done before the XY tilts above.

Align Receiver to Chief Ray   If unchecked, the fiber is positioned at XY coordinates (0.0, 0.0) on the image surface. If checked, the fiber is repositioned in x and y to be centered at the point the chief ray intercepts the image surface. In either case the receiving fiber is oriented along the image space z axis, unless rotated using the controls above.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Use Huygens Integral   If checked, the Huygens PSF is used for the beam mode at the receiving fiber. Computing the Huygens PSF is much slower than the direct ray-based integration used if this option is unchecked. However, there are cases where the full Huygens PSF is required. These cases generally have severe caustics or discontinuities in the wavefront. The recommended approach is to leave this option unchecked. If the fast ray- based integration cannot be computed accurately, the fiber coupling will be zero and a warning will be printed which will recommend switching to the Huygens algorithm.

Discussion

This feature computes fiber coupling for single-mode fibers with a Gaussian shaped mode. For multi-mode fiber coupling, see "Calculating efficiency of multi-mode fibers". For a more detailed and flexible physical optics treatment of fiber coupling see "Computing Fiber Coupling".

Fiber coupling efficiency is computed based upon a two fiber or a one fiber model. In the two fiber model, light emerges from a source fiber to fill or overfill the entrance pupil of an optical system. Energy not collected by the entrance pupil is lost, reducing the overall efficiency. If desired, the source fiber may be ignored for a one fiber model; and in this case the efficiency is computed relative to the energy entering the entrance pupil; which in turn is a function of the system apodization (see "Apodization Type").

The system efficiency (S) is the sum of the energy collected by the entrance pupil which passes through the optical system, accounting for both the vignetting and transmission of the optics (if polarization is used), divided by the sum of all the energy which radiates from the source fiber:

$$S = \left[\frac{\iint_E t(x,y)F_s(x,y)dxdy}{\iint F_s(x,y)dxdy} \right]^2$$

where F_s is the source fiber amplitude function and the integral in the numerator is only done over the entrance pupil of the optical system, and $t(x,y)$ is the amplitude transmission function of the optics. The transmission is affected by bulk

absorption and optical coatings (if use polarization is checked on). If the source fiber is ignored, then the integral in the denominator is only done over the entrance pupil, and the F_s function is determined by the system apodization, if any.

Aberrations in the optical system introduce phase errors which will affect the coupling into the fiber. Maximum coupling efficiency is achieved when the mode of the wavefront converging towards the receiving fiber perfectly matches the mode of the fiber in both amplitude and phase at all points in the wavefront. This is defined mathematically as a normalized overlap integral between the fiber and wavefront amplitude:

$$T = \frac{|\iint F_r(x,y)W'(x,y)dxdy|^2}{\iint F_r(x,y)F_r'(x,y)dxdy \iint W(x,y)W'(x,y)dxdy}$$

where $F_r(x,y)$ is the function describing the receiving fiber complex amplitude, $W(x,y)$ is the function describing the complex amplitude of the wavefront from the exit pupil of the optical system, and the ' symbol represents complex conjugate. Note that these functions are complex valued, so this is a coherent overlap integral.

T has a maximum possible value of 1.0, and will decrease if there is any mismatch between the fiber amplitude and phase and the wavefront amplitude and phase.

Zemax computes the values S and T. The total power coupling efficiency is the product of these numbers. A theoretical maximum coupling efficiency is also computed; this value is based upon ignoring the aberrations but accounting for all vignetting, transmission, and other amplitude mismatches between the modes.

To model the reflection loss from the front face of the receiving fiber, set the image space glass to the material the fiber is made of, and check the box "use polarization". The reflectivity of the receiving fiber (and other surfaces) will be accounted for.

Multi-Mode Coupling

Compute the coupling efficiency of the optical system into a multi-mode fiber of a specified NA and radial aperture by using the NA setting on the Geometric Image Analysis feature.

Zemax has an algorithm for accurately computing fiber coupling into single-mode fibers; for details see "Fiber Coupling Efficiency". Also see "Computing Fiber Coupling".

To estimate the coupling efficiency for multi-mode fibers, a geometric approach may be used. Place a circular aperture at or just before the image surface with the appropriate maximum radial aperture representing the core size. Then set the NA (see the table above) to the maximum acceptance NA of the fiber. The percent efficiency will then be calculated by summing all the rays that pass the core aperture within the specified NA. The NA of a typical multi-mode fiber with an inner core of index n_i and an outer cladding of index n_o is given by

$$NA = \sqrt{n_i{}^2 - n_o{}^2}$$

Polarization and Surface Physics Group

Polarization

For more information on polarization calculations, see "POLARIZATION ANALYSIS" in the manual.

Polarization Ray Trace

The polarization ray trace feature generates a text window which displays all the polarization data for a single ray.
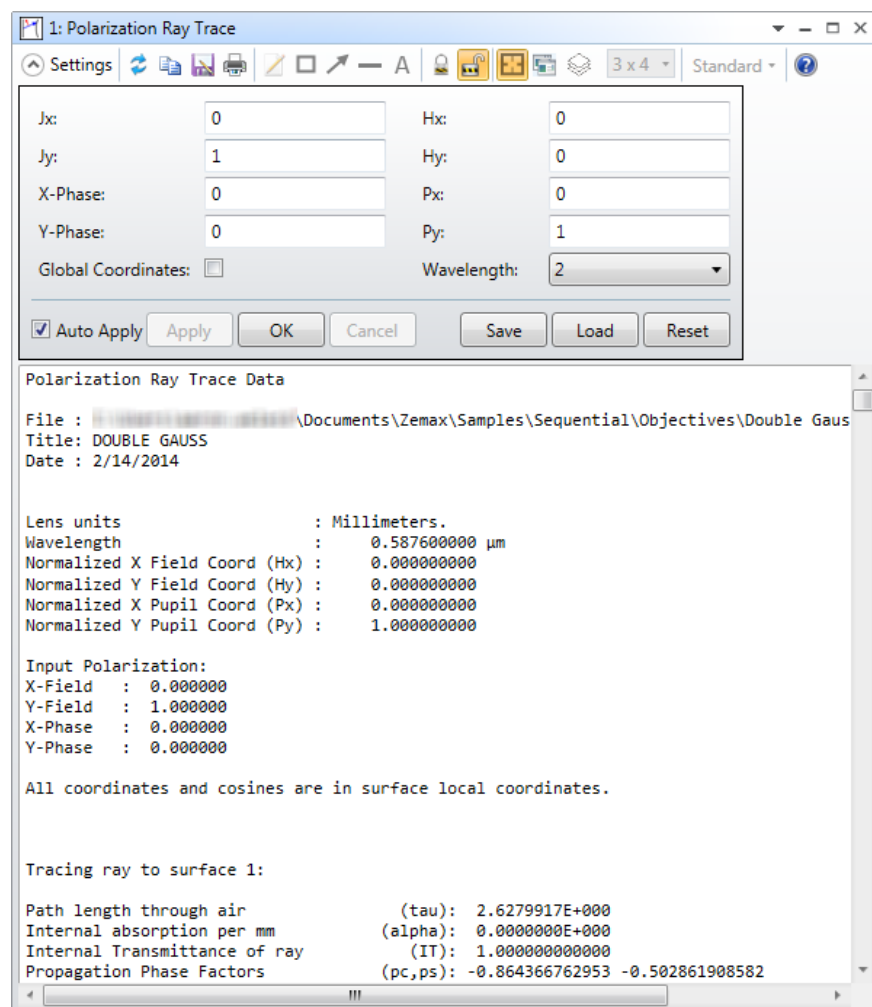
Jx, Jy   Jones electric field. see "Defining the initial polarization" .

X, Y Phase   Phase of the X, Y component of the Jones electric field in degrees.

Hx   Normalized x-field coordinate. The value should be between -1 and 1.

Hy   Normalized y-field coordinate. The value should be between -1 and 1.

Px   Normalized x-pupil coordinate. The value should be between -1 and 1.

Py   Normalized y-pupil coordinate. The value should be between -1 and 1.

Wavelength   The wavelength number of the ray to trace.

Global Coordinates   If checked, all coordinate and cosine data are given in global rather than local coordinates.

Discussion This feature tabulates all of the data computed by Zemax to perform polarization ray tracing. See "POLARIZATION ANALYSIS".

Polarization Pupil Map

Generates a graph of the polarization ellipse as a function of pupil position. This aids in visualizing the change in polarization over the pupil. This feature may also be used to compute transmission in systems modeled using interfering paths in separate configurations, such as birefringent polarizing beam splitters and interferometers.
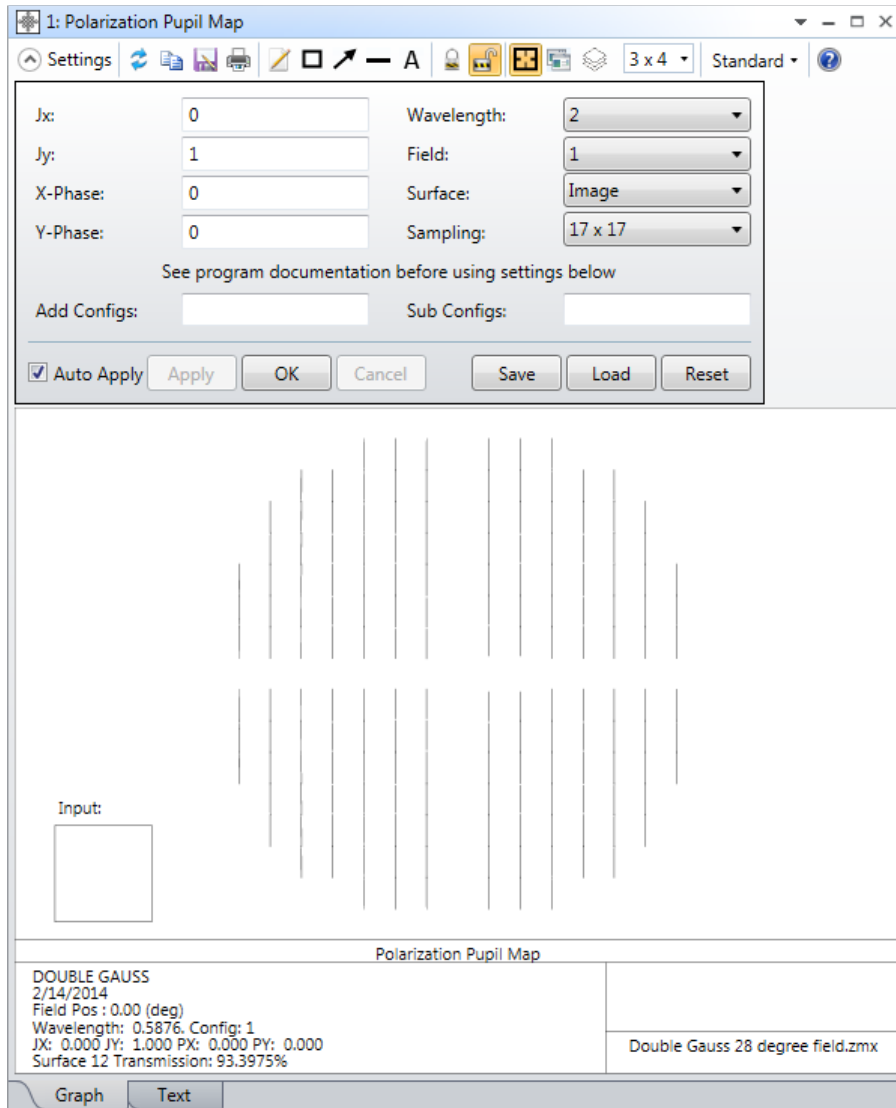
Jx, Jy   Jones electric field. see "Defining the initial polarization".

X, Y Phase   Phase of the X, Y component of the Jones electric field in degrees.

Wavelength   The wavelength number of the rays to trace.

Field   The field position number of the rays to trace.

Surface The surface at which the data is shown. Data is after refraction through the specified surface.

Sampling   The grid size of the sampling in the pupil.

Add Configs

Sub Configs   If the "Add Configs" and "Sub Configs" settings are left blank, the polarization ellipse and transmission is computed for a single configuration (selected in the drop-down menu at the top of the window; "Current" is the default setting) and no interference effects are considered. To model interference between configurations see the discussion below.

Discussion

The polarization ellipse is a representation of the figure traced out by the electric field vector as the wave propagates during one cycle. The magnitude of the ellipse is determined by the transmission of the ray, which is generally a function of pupil position. See "POLARIZATION ANALYSIS".

The text of the data computed by this feature lists the Px, Py, Ex, Ey, Intensity, Phase, and Orientation values. These are the normalized pupil x and y coordinates, electric field magnitude x and y components, the electric field intensity ($Ex^2 + Ey^2$), the phase difference between the Ex and Ey phase in degrees, and the orientation angle of the major axis of the polarization ellipse in degrees. Note Ex and Ey are both complex and the values listed here are the magnitude of the amplitude, or sqrt(E*E).

Modeling interference between configurations

Some optical systems require more than one configuration to model the complete optical beam. The most common example is a polarizing beamsplitter that uses uniaxial crystals to separate orthogonal polarizations of the incident electric field. Typically, these beamsplitters implement 2 crystals with orthogonal axis orientations,

each requiring an ordinary and an extraordinary ray trace configuration, and therefore these systems require a total of 4 configurations to model all possible ray paths.

Computing transmission through these systems can be complicated. If a single incident ray is decomposed into 4 individual rays, one for each configuration, each ray will carry a portion of the electric field and energy. It is a fundamental property of rays that the ray trace does not depend upon other rays in other configurations. Thus, if each ray is propagated individually, each will carry some portion of the electric field. If in two or more configurations the ray paths are superimposed, then the coherent sum of the individual ray paths must be considered. It is possible that two or more configurations create electric fields of comparable magnitude but opposing phase, resulting in destructive interference. This destructive interference results in a lower total transmission for these rays. From a physical optics viewpoint, this destructive interference means the energy could never have propagated along these rays in the first place, and must have propagated elsewhere. There is no practical, general way for Zemax to compute where this energy actually went.

To address this difficulty, the "Add Configs" and "Sub Configs" allow user specification of which configurations should be coherently summed. The "Add" configurations are all ray traced, and the electric fields are coherently summed. The "Sub" configurations are also all ray traced, and these are separately coherently summed. As part of the coherent sum, Zemax can determine how much energy was lost due to destructive interference between the rays in the "Sub" configurations. This energy is then added to the "Add" configurations, and the total transmission is then determined. To list multiple configurations, separate the configuration numbers with a blank space. For example, to define configurations 1 and 2 as the "Add" configurations enter "1 2".

This feature is particularly useful for analysis of birefringent systems, where two or more configurations (ordinary and extraordinary) are required to trace all possible beam paths. The computations may be applicable to other systems as well, but great care should be taken before any computed results are trusted. This feature may yield meaningless or misleading results if the "Add" and "Sub" configurations are defined in a way that does not reflect how the beams actually interfere with one another.

Transmission

Computes the integrated and surface by surface transmission through the optical system considering polarization effects.
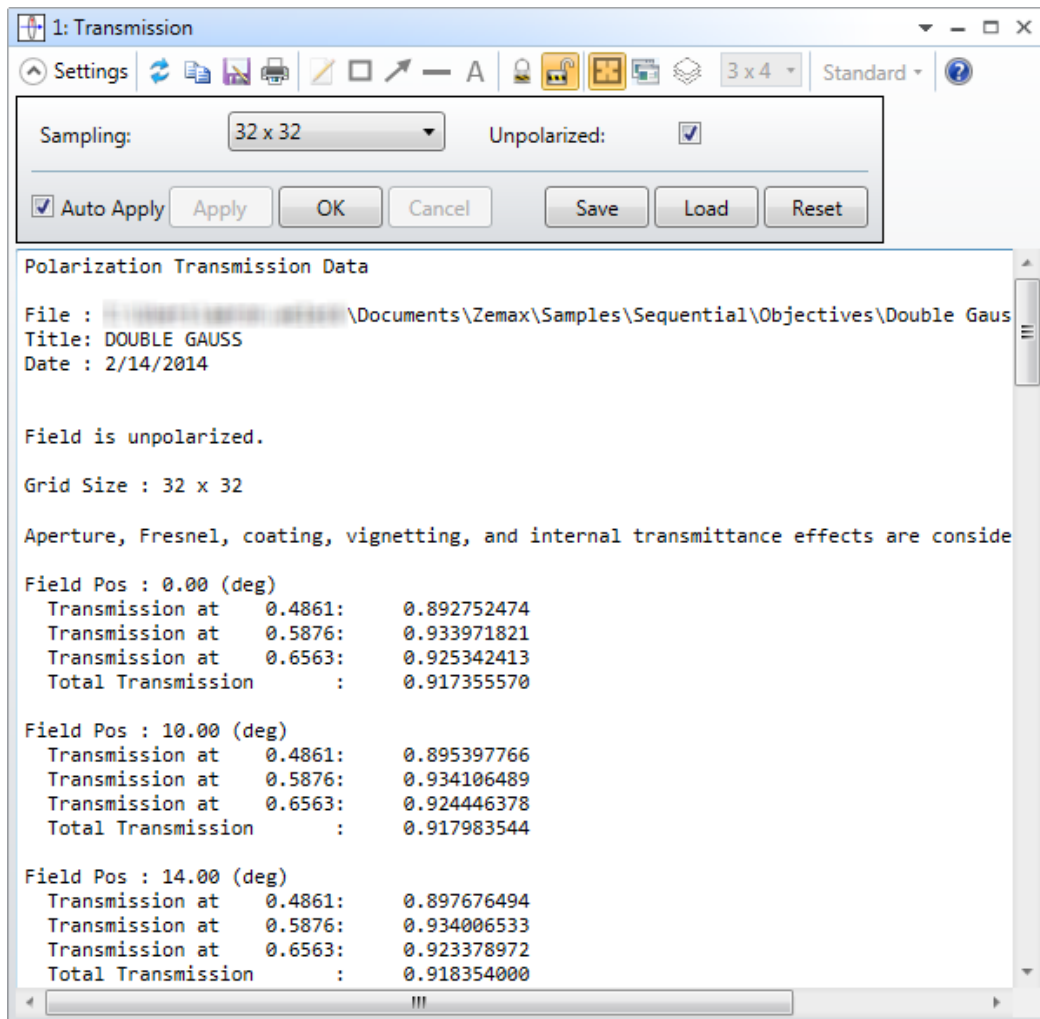
Jx, Jy Jones electric field. see "Defining the initial polarization".

X, Y Phase Phase of the X, Y component of the Jones electric field in degrees.

Sampling Specify the density in the pupil. Larger grid sizes are more accurate, although the computation time increases.

Unpolarized If checked, the electric field definition is ignored and the unpolarized computation is performed.

Discussion

This feature tabulates for each field position and wavelength the integrated transmission of the optical system for the specified polarization. The transmission is computed as a fraction of 100%, with 100% being that transmitted if there were no absorption, reflection, or vignetting losses. The transmission calculation accounts for vignetting factors, vignetting due to apertures or obscurations, ray clipping due to ray trace errors, surface Fresnel or coating losses, and bulk internal transmittance due to absorption.

Also tabulated for each field and wavelength is the relative and total transmission of the chief ray. This allows identification of where significant surface losses originate.

See also "Relative Illumination".

Phase Aberration

Computes the polarization induced phase aberration of an optical system.
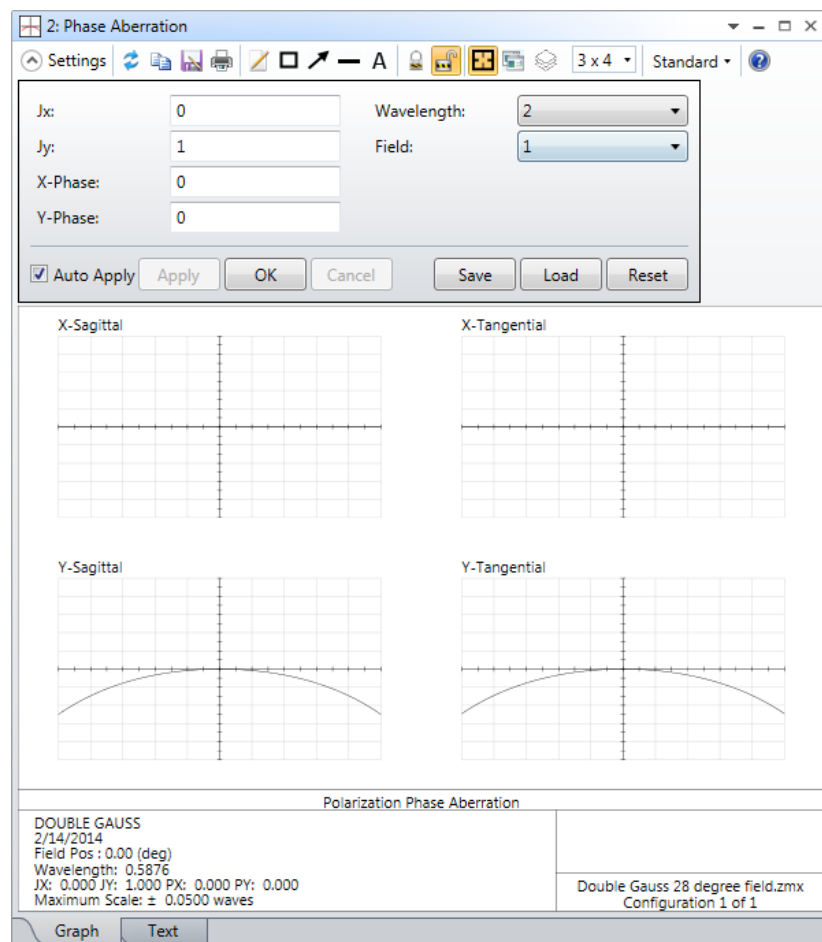
Jx, Jy   Jones electric field. see "Defining the initial polarization".

X, Y Phase   Phase of the X, Y component of the Jones electric field in degrees.

Wavelength   The wavelength number of the rays to trace.

Field   The field position number of the rays to trace.

Discussion

Polarization phase aberrations are induced by the effects of refraction through dielectric media, and by reflection from metallic or dielectric mirrors. This feature computes for the specified field position and wavelength the polarization phase aberration in the image space for the X and Y orientations of the electric field vector as a function of entrance pupil coordinate. The aberration is defined as the variation in the phase of the electric field in one direction, such as X or Y, as a function of the pupil position. For example, if the electric field in the X direction is $(0.7 + 0.7i)$ for the chief ray (note the E field values are complex), and at some other point in the pupil the electric field in the X direction is $(-0.7 + 0.7i)$ then the polarization phase aberration between these points is one-quarter wave (the phase of Ex changed from 45 degrees to 135 degrees, or one-quarter wave). Note this is completely distinct from the phase differences between the Ex

and Ey fields, which describe the polarization state (linear or circular for example). For phase differences between Ex and Ey see "Power Pupil Map".

Like the usual OPD plots, the polarization phase aberration is referenced to the chief ray. However, there are cases where the chief ray phase cannot be determined in both orientations. For example, in an axial symmetric system, if the incident polarization is linear in the Y direction, there is zero intensity in the X direction for the chief ray, and therefore the X phase is indeterminate. For other rays in the pupil, there is generally a slight rotation of the polarization, and therefore the resulting electric field in the X direction yields a valid phase angle. To avoid this discontinuity in the phase, Zemax averages two rays on either side of the chief ray to interpolate the chief ray phase. The problem may still appear under certain circumstances, even with this averaging technique. In all cases, the phase data is still valid, because the phase aberration has no effect on image quality if the intensity is zero.

Transmission Fan

Generates a graph of the transmitted intensity for each field and wavelength as a function of either tangential or sagittal pupil fans.
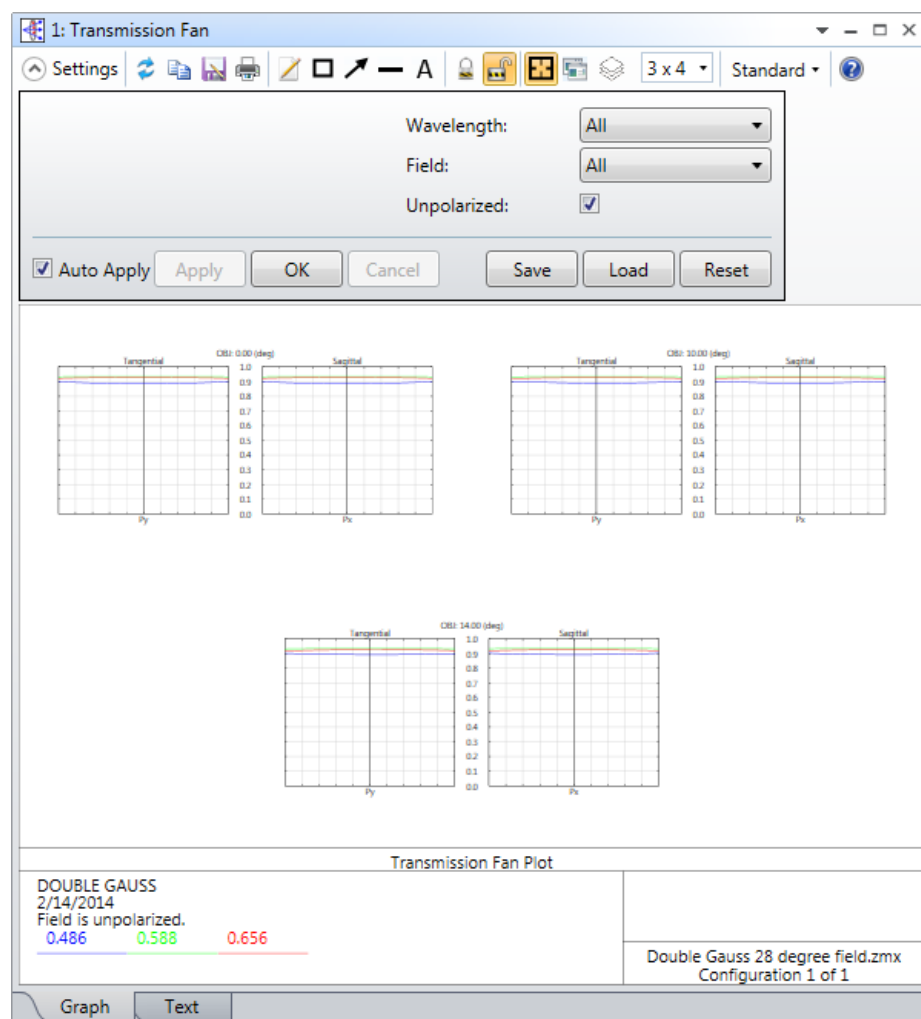
Jx, Jy Jones electric field. see "Defining the initial polarization".

X, Y Phase Phase of the X, Y component of the Jones electric field in degrees.

Wavelength   The wavelength number of the rays to trace.

Field   The field position number of the rays to trace.

Unpolarized   If checked, the electric field definition is ignored and the unpolarized computation is performed.

Discussion The polarization transmission fan is useful for determining the transmission variation over the pupil as a function of field and wavelength. See "POLARIZATION ANALYSIS".

Surface
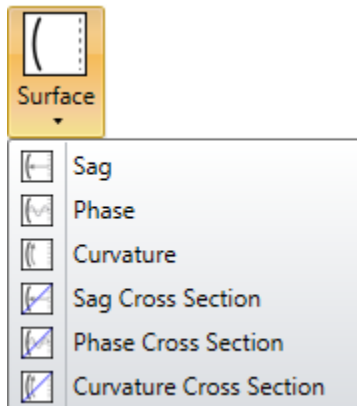
Surface Sag

Displays the sag of a surface as a 2D color or contour map, or as a 3D surface plot.
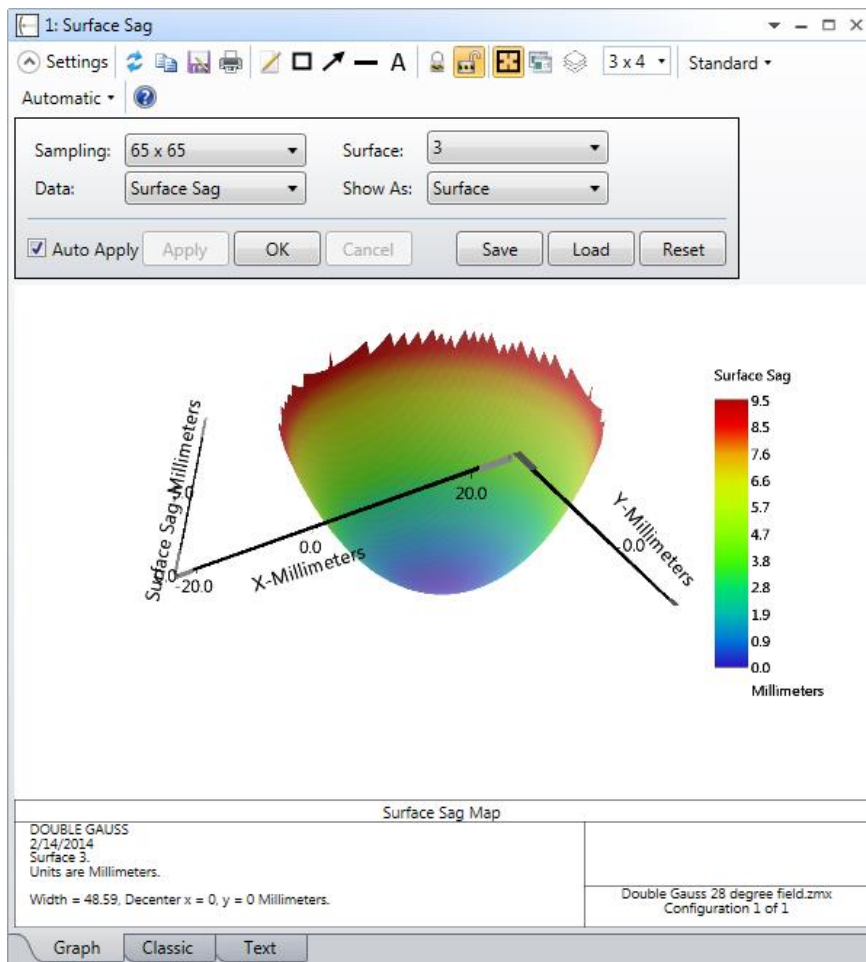
Sampling   The size of the grid used. The sampling may be 33x33, 65x65, etc.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String" below. The contours are defined in lens units.

Surface The surface number to compute the sag display for.

Data   Choose surface sag.

Show As   Choose surface plot, contour map, grey scale, or false color map as the display option.

Discussion:This feature accounts for the size and shape of any aperture present on the surface; even if the aperture is decentered. The sag is computed on a uniform grid of points in XY plane, and the Z value of the sag is the displayed data.

See also the Surface Phase feature described below.

The Contour Format String

The Contour Format string allows some control over the appearance of the contour map.

If left blank, the default contour spacing will be selected. If a single value is entered, then the contour spacing will be set to this value. For example, if 0.05 is entered, the contour interval will be 0.05. If multiple values separated by spaces are entered, only those contours will be drawn. For example, if "0.8 0.5 0.2" is entered, only those three contours will be

drawn. If the specified contour settings result in too many contours to be drawn properly, the default number of contours will be selected.

Surface Phase

Displays the phase of a surface as a 2D color or contour map, or as a 3D surface plot.
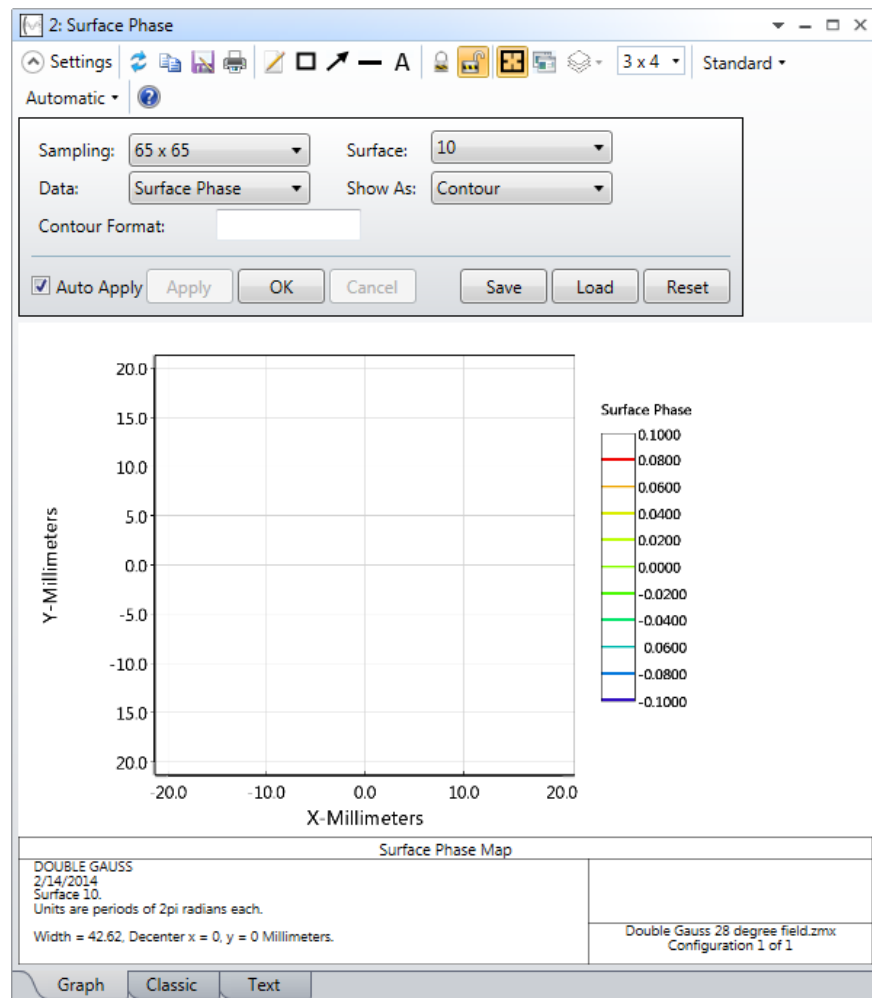
Sampling   The size of the grid used. The sampling may be 33x33, 65x65, etc.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String". The contours are defined in periods; each period is a phase change of 2π .

Surface The surface number to compute the phase display for.

Data   Choose surface phase.

Show As   Choose surface plot, contour map, grey scale, or false color map as the display option.

Discussion This feature accounts for the size and shape of any aperture present on the surface; even if the aperture is decentered. The phase is computed on a uniform grid of points in XY plane, and the phase value is the displayed data.

This feature defines phase in units of periods; so one period represents a phase change of 2π. Surfaces which do not impart a phase change to the ray, such as the Standard surface, will display a phase of zero everywhere on the surface phase display. See also the Surface Sag feature described above.

Surface Curvature

Displays the tangential, sagittal, x, or y curvature of a surface as a 2D color or contour map, or as a 3D surface plot.
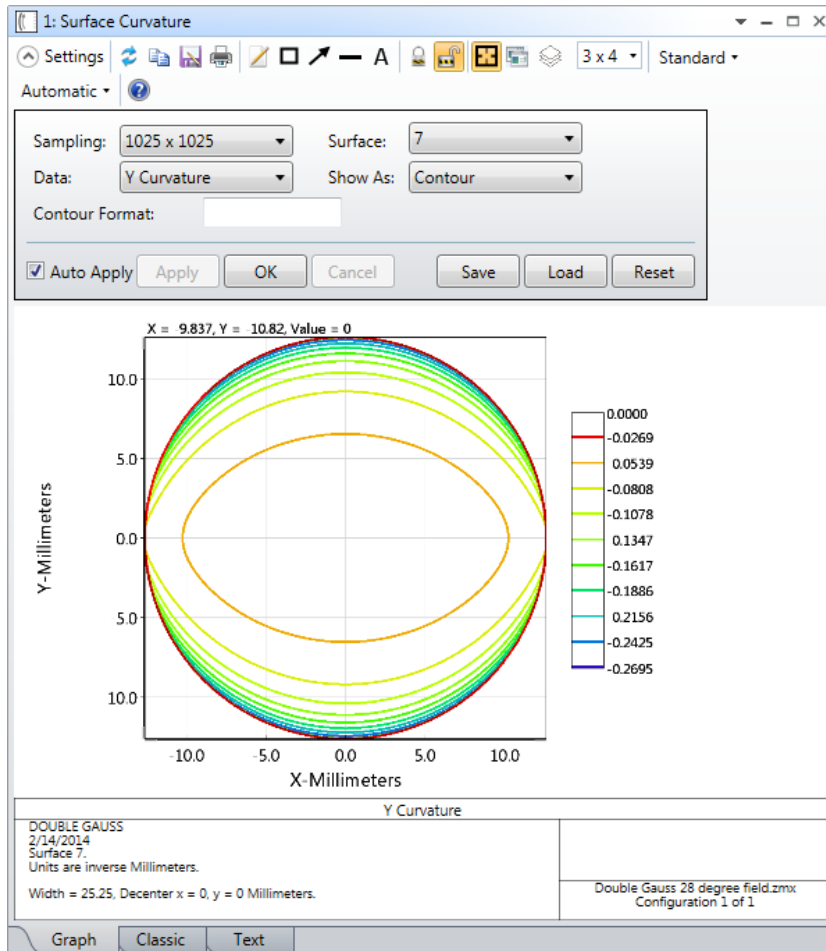
Sampling The size of the grid used. The sampling may be 33x33, 65x65, etc.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String". The contours are defined in lens units.

Surface The surface number to compute the sag display for.

Data Choose tangential. sagittal, x, or y curvature.

Show As Choose surface plot, contour map, grey scale, or false color map as the display option.

Discussion This feature accounts for the size and shape of any aperture present on the surface; even if the aperture is decentered.

Surface Sag Cross Section

Displays the sag of a surface as a cross section.
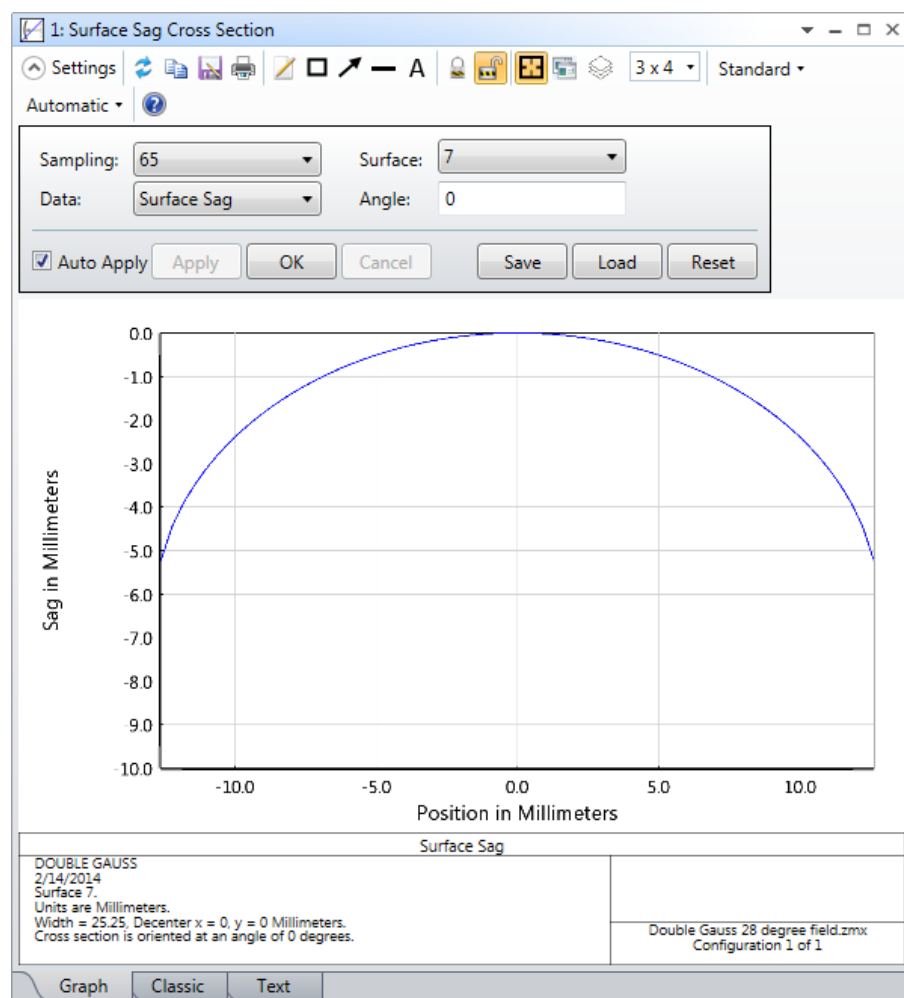
Sampling   The sampling may be 33, 65, etc.

Surface The surface number to compute the sag display for.

Data   Choose surface sag.

Angle   Choose the orientation angle in degrees relative to the local x axis for the cross section.

Discussion This feature calculates the surface sag as a cross section across the full diameter of the surface. See also the Surface Phase Cross Section feature described below.

Surface Phase Cross Section

Displays the phase of a surface as a cross section.
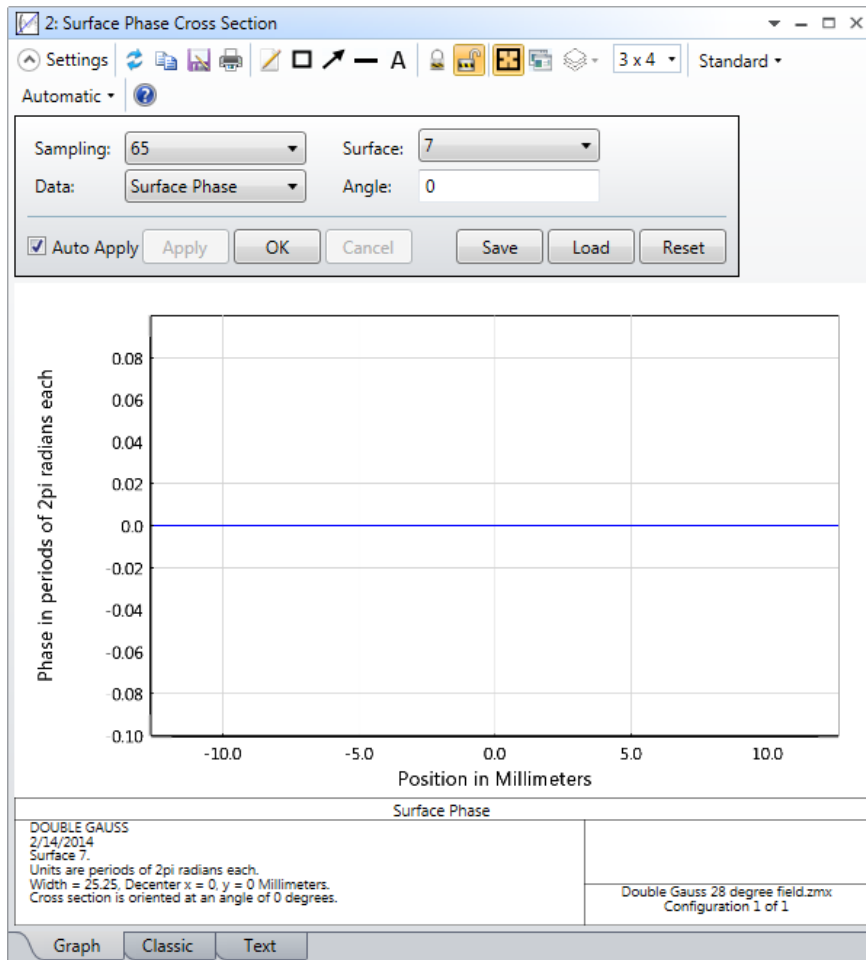
Surface The surface number to compute the phase display for.

Data Choose surface phase.

Angle Choose the orientation angle in degrees relative to the local x axis for the cross section.

Discussion This feature calculates the surface phase as a cross section across the full diameter of the surface. This feature defines phase in units of periods; so one period represents a phase change of 2π . Surfaces which do not impart a phase change to the ray, such as the Standard surface, will display a phase of zero everywhere on the surface phase display. See also the Surface Sag Cross Section feature described above.

Surface Curvature Cross Section

Displays the curvature of a surface as a cross section.
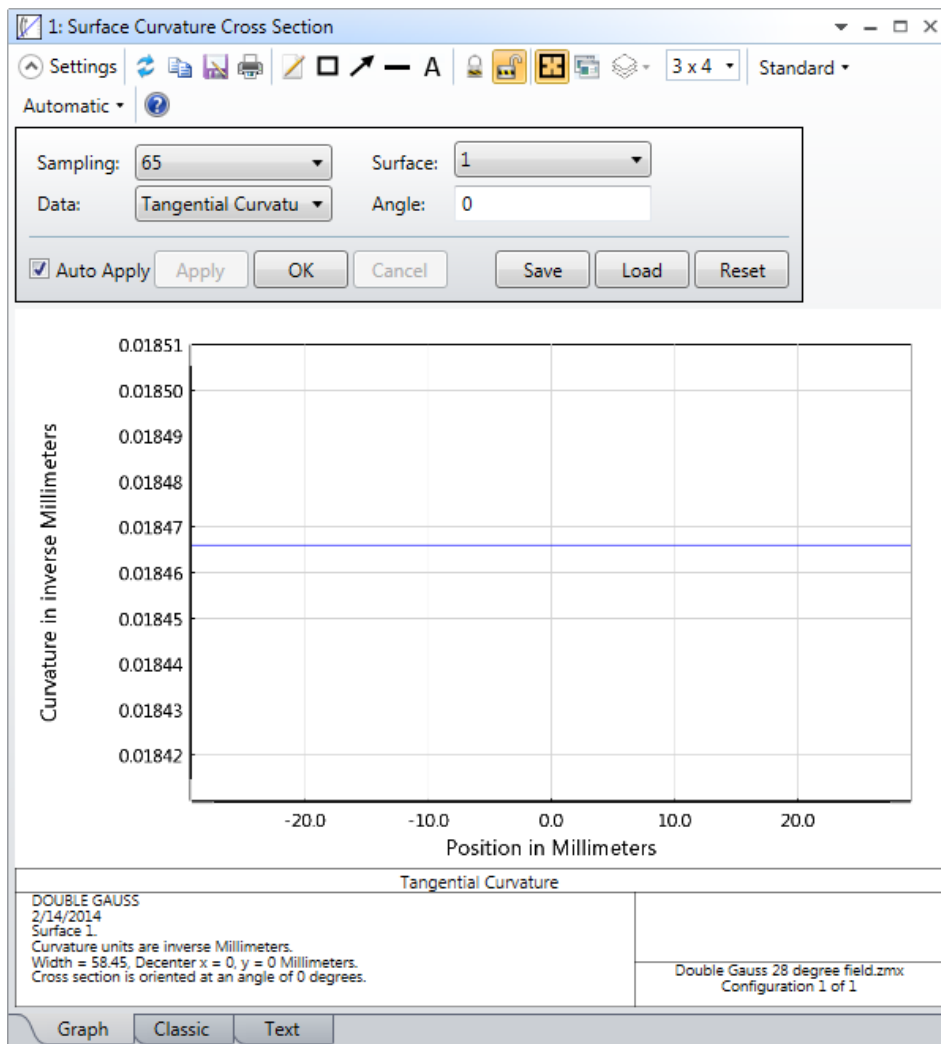
Sampling The sampling may be 33, 65, etc.

Surface The surface number to compute the sag display for.

Data Choose tangential, sagittal, x, y, tangential+sagittal, or x+y curvature.

Angle Choose the orientation angle in degrees relative to the local x axis for the cross section.

Discussion This feature calculates the surface curvature cross section across the full diameter of the surface. See also the Surface Sag Cross Section feature described above.

Coatings
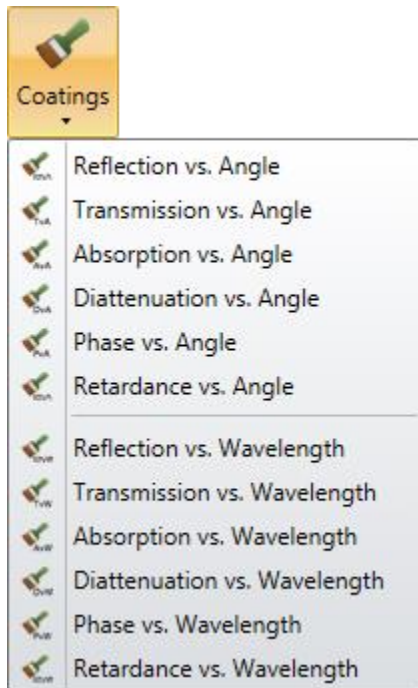
For more information on coating calculations, see "POLARIZATION ANALYSIS".

Reflection vs. Angle

Computes the S, P, and average polarization intensity coefficients for reflection for the specified surface as a function of incident angle.

Min Angle The minimum angle of incidence to plot. This defines the left edge of the plot.

Max Angle The maximum angle of incidence to plot. This defines the right edge of the plot.

Min Y The minimum y value to plot. This defines the bottom edge of the plot.

Max Y The maximum y value to plot. This defines the top edge of the plot.

Surface The surface number which defines the interface to use.

Object The object number if the surface is a Non-Sequential Components surface.

Face The face number which defines the interface to use, if the surface is a Non-Sequential Components surface.

Wavelength The wavelength number to use.

Reverse Direction If the selected surface is a Non-Sequential Components surface, choosing this option will reverse the incident media and substrate.

Discussion The incident angle is measured in the medium prior to the specified surface.

Transmission vs. Angle

Computes the S, P and average polarization intensity coefficients for transmission for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Absorption vs. Angle

Computes the S, P, and average polarization intensity coefficients for absorption for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Diattenuation vs. Angle

Computes the R (reflected) and T (transmitted) diattenuation for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Phase vs. Angle

For sequential surfaces, this feature computes the S and P polarization phase for reflection (if the glass is MIRROR) or for transmission (if the glass is not MIRROR) as a function of incident angle. For non-sequential objects, this feature computes the S and P polarization phase for reflection (if the material is MIRROR) or for transmission (if the material is not MIRROR) as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Retardance vs. Angle

Computes the retardance for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Reflectivity vs. Wavelength

Computes the S, P, and average polarization intensity coefficients for reflection for the specified surface as a function of incident wavelength.

Min Wave The minimum wavelength to plot. This defines the left edge of the plot.

Max Wave The maximum wavelength to plot. This defines the right edge of the plot.

Min Y The minimum y value to plot. This defines the bottom edge of the plot.

Max Y The maximum y value to plot. This defines the top edge of the plot.

Surface The surface number which defines the interface to use.

Object The object number if the surface is a Non-Sequential Components surface.

Face The face number which defines the interface to use, if the surface is a Non-Sequential Components surface.

Angle The angle of incidence, in degrees, to use.

Reverse Direction If the selected surface is a Non-Sequential Components surface, choosing this option will reverse the incident media and substrate.

Discussion See "POLARIZATION ANALYSIS".

Transmission vs. Wavelength

Computes the S, P, and average polarization intensity coefficients for transmission for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Discussion See "POLARIZATION ANALYSIS".

Absorption vs. Wavelength

Computes the S, P, and average polarization intensity coefficients for absorption for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Diattenuation vs. Wavelength

Computes the R (reflected) and T (transmitted) diattenuation for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Phase vs. Wavelength

For sequential surfaces, this feature computes the S and P polarization phase for reflection (if the glass is MIRROR) or for transmission (if the glass is not MIRROR) as a function of incident wavelength. For non- sequential objects, this feature computes the S and P polarization phase for reflection (if the material is MIRROR) or for transmission (if the material is not MIRROR) as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Retardance vs. Wavelength

Computes the retardance for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Reports

Surface Data

Display surface data.
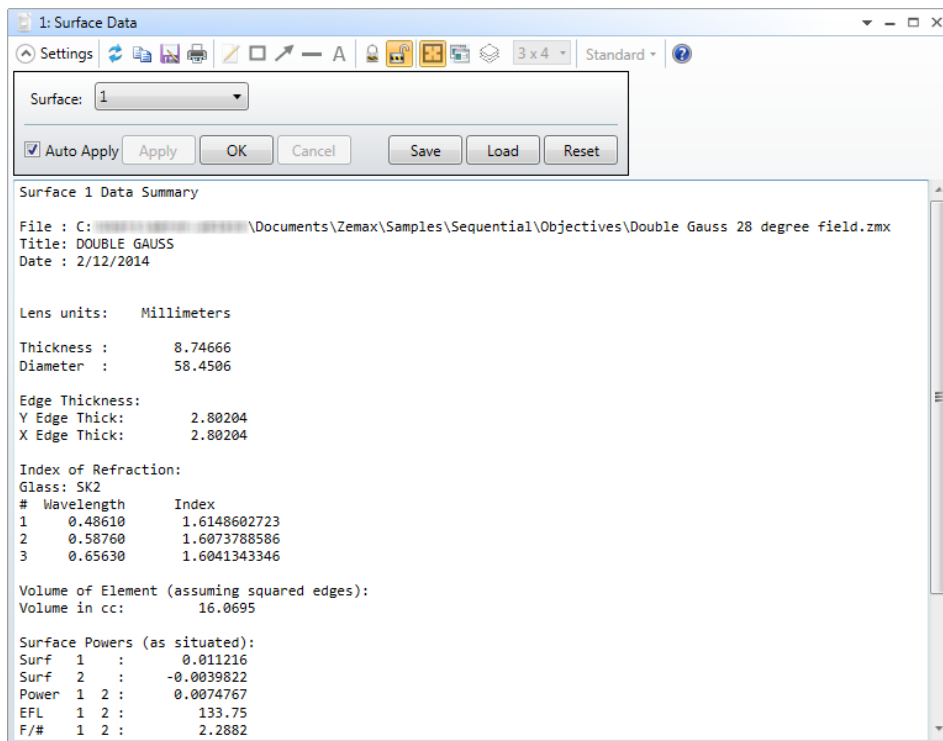
Surface The surface number to display the data for.

Discussion

This feature generates a text window which displays surface-specific data. The data includes surface and element powers and focal lengths, edge thickness, index of refraction, and other data for the surface.

If the glass type of the surface is a "model" glass, then Zemax will list out the index of refraction at each defined wavelength calculated from the model glass parameters. Also listed is the name of the "Best Fit Glass", which is the name of the glass in the currently loaded catalog(s) which has the closest index of refraction to the model glass in an RMS sense. Specifically, Zemax computes the index error as the sum of the squares of the difference between the model index and the actual glass index using the dispersion formulas. The sum is over the defined wavelengths. The index error is computed for every glass in the current catalogs, and the glass with the lowest RMS index deviation is designated as the best fit glass.

Note that the best fit glass may have a different V number than the model glass, however, this is due to the approximations made in the model glass dispersion. Since index of refraction is the physically significant parameter, only the index is used is making the glass selection. When changing from model glass to a real "fixed" glass, the same algorithm is used to make the selection. For more information on model glasses, see "Using model glasses".

System Data

Display system data.
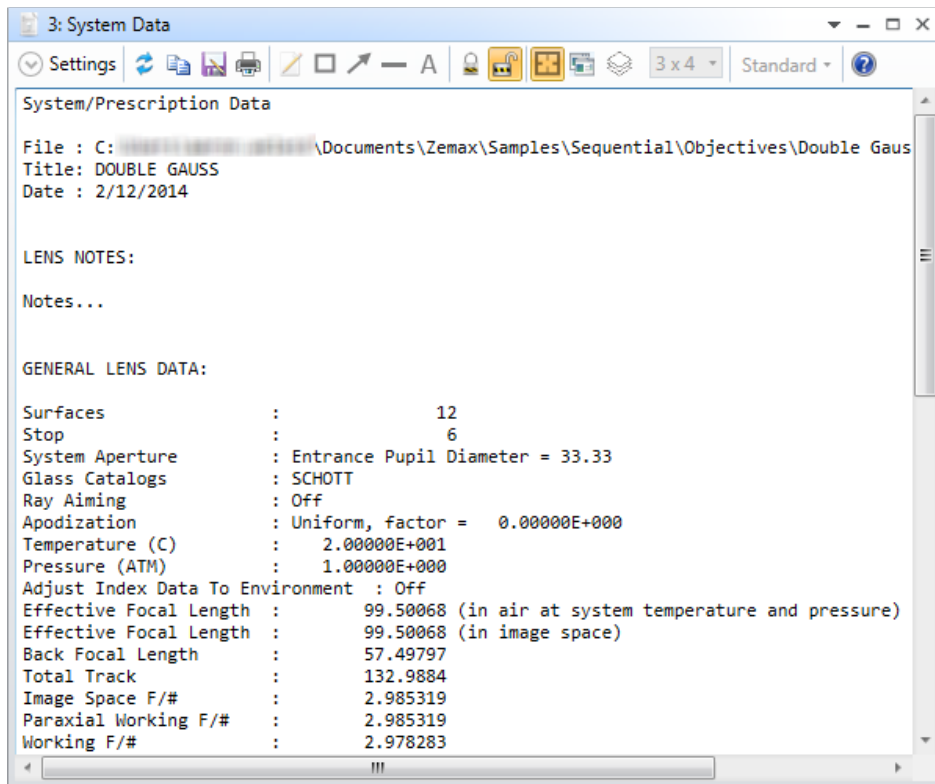
Discussion This feature generates a text window which lists many system-related parameters, such as pupil positions and sizes, magnification, F/#, etc.

Prescription Data

This function generates a list of all surface data, and summarizes the lens system. This is the feature to use for printing the contents of the Lens Data Editor.

General Data   Includes F/#, pupil positions, magnification, etc.

Surface Data   Surface type, radii, thickness, glass, semi-diameter, conic.

Surface Detail   Surface Parameter data. All NSC object data is also listed in this section for NSC groups.

Edge Thickness The X and Y edge thickness of each surface.

Multi-Config Data   A table of multi-configuration operand data.

Solves/Variables Solve types and data, variables.

Index Data   Index of refraction and TCE data for each wavelength/surface.

Global Vertex   Global coordinates of the vertex and the rotation matrix for each surface. See "Global Coordinate Reference Surface" for a discussion of this data.

Center Of Curvature   Global coordinates of the center of curvature of rotationally symmetric surfaces. The radius, position, and orientation of the surface are used to determine the point at the vertex center of curvature. The location of this point is useful for alignment.

Element Volume Volume in cc, density, and mass. See below.

F/ Numbers Lists working F/# for each field position and wavelength combination.

Cardinal Points Lists the locations of focal, principal, anti-principal, nodal, and anti-nodal points.

POP Settings Lists the surface by surface settings used by the Physical Optics Propagation feature.

Files Used Lists all the files used by the optical system model, including glass data, macros, extensions, CAD files, DLL files, and others.

Discussion This report lists specification data, indices of refraction, global coordinates, element volumes, and more. It is suitable for describing the lens prescription.

Comments on computing element volumes

When Zemax computes element volumes for spherical or plane standard surfaces with circular edges, the edge of the surface with the smaller diameter is assumed to be "squared up" to the semi-diameter of the larger surface. The volume is exact for this common special case.

For other elements, with possibly aspheric faces, elliptical, annular, or rectangular apertures and/or decentered apertures, a numerical integration technique is used that yields approximate volumes to an accuracy of about

0.1%. If the surface is rotationally symmetric, the edge of the surface with the smaller diameter is "squared up" to the semi-diameter of the larger surface. Otherwise, the edge of the surface with the smaller diameter is "tapered" to the semi-diameter of the larger surface. If zero volume is reported than the algorithm is unable to compute the volume to sufficient accuracy. This happens when the aperture types are not the same, or have different sizes or decenters.

When computing the density of elements, the density in grams per cubic centimeter for a catalog glass is retrieved from the glass catalog. For gradient index surfaces, Zemax assumes the density is 3.6 g/cc, which may not be a good estimate.

System Summary Graphic

Displays in a graphic window a summary of the system data, similar to the text based system data report.
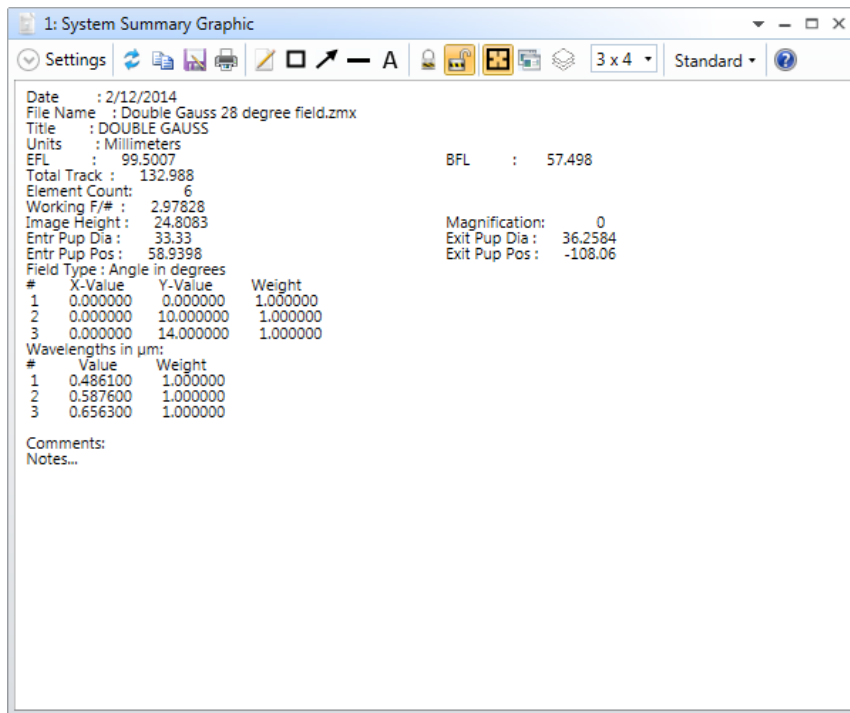
Discussion This graphic is primarily used to display a summary of system data within the printed page of the report graphics 4 or 6 feature; described in "Report Graphics 4/6".

Cardinal Points

This feature lists for the selected range of surfaces and wavelengths the locations of the focal, principal, anti- principal, nodal, anti-nodal planes. The calculation is done for any defined wavelength and either the X-Z or Y-Z orientation.

Cardinal Point Data settings and summary window

First Surface	The starting surface number of the group to compute the cardinal points for.
Last Surface	The ending surface number of the group to compute the cardinal points for.
Wavelength	The wavelength number to use for the computation.
Orientation	The orientation to use for computing the cardinal plane locations.

Discussion This feature may not return reliable results if coordinate breaks or non-centered optics are included within the specified surface range.

Universal Plot Group

This feature is only available in the Professional and Premium editions of Zemax.

New Universal Plot 1D...

Displays as either a plot or as a text listing the value of any optimization operand as a function of one other parameter. See also "New Universal Plot 2D...".

dependent Variable X Settings

Select either surface, system, multi-configuration, or NSC data as the independent variable.

Select the parameter to use as the independent variable. The available selections include radius, curvature, thickness, conic, parameter, and extra data values if surface parameters are chosen. For system parameters, the selections include the system aperture, field and wavelength data, apodization factor, temperature, and pressure. For configuration data, all multi-configuration operands are listed. For NSC data, object position and parameter data are listed.

The surface number for the independent variable parameter if the parameter is associated with a surface. This field is used for the configuration number if configuration data is selected. If NSC data is selected, this field is used for the object number.

Start/Stop Value The beginning and ending range of the independent variable.

# of Steps The number of values between the start and stop values, inclusive, to compute the dependent variable function value.

Dependent Variable Y Settings

Operand The optimization operand to use as the dependent variable function. The current system merit function is also included as "Merit". If the system merit function is selected, then either the entire merit function or any specific operand's value may be selected using the "Line" control below.

Line The optimization operand # to use as the dependent variable function if "Merit" is selected for the Operand.

Int1/Int2 The Int1 and Int2 values for the selected operand. The dialog box will display the short name for the Int1 and Int2 values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Data1-Data4 The Data values for the selected operand. The dialog box will display the column heading name for these values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Min/Max Plot Value The min and max value to plot for the dependent variable (Y) function. If both of these values are zero, then a default scale will be chosen. If either one is not zero, then the y range of the plot will extend from the minimum to the maximum. If the data does not fit within the specified range, then the data may plot beyond the borders of the data box.

Plot Title Any text may be entered which will be used as the title for the plot or text listing.

Save As New Universal Plot This will save the current settings in a Universal Plot 1D (UPL) file specified by the user. Any name which may be used as a valid file name may be entered. Although the "browse" window allows the file to be saved in any folder, Zemax will only recognize the file if it resides in the <data>\Miscellaneous folder (see "Folders").

Once the settings are saved, the file name will appear as a menu option under the Universal Plot 1D menu option for quick regeneration of the plot.

Load From This will load the settings from a previously saved Universal Plot 1D (UPL) file specified by the drop down list to the right of the button.

Discussion

This feature will either plot as a graph or create a text listing of the value of any optimization operand as a function of a single system, surface, or multi-configuration value.

For example, suppose a plot of the sagittal MTF at 30 lp/mm was needed as a function of the decenter of a lens group (which may be a useful diagnostic for tolerancing analysis). Since the MTFS operand computes the sagittal MTF, the Universal Plot can generate such a graphic or text listing. See "Optimization operands" for a list of available optimization operands. A decenter of a lens group is defined by Parameter 1 or 2 on the relevant coordinate break surface, and Parameter 1 and 2 are both listed in the available surface group parameters.

Because of the number of different plots this feature can generate, there are no intelligent defaults for either the independent or dependent settings. These values need to be carefully provided in the settings dialog box. If the optimization operand cannot be computed, an error message will be displayed and the plot will not be generated.

Because many optimization operands accept Hx and Hy values to define the field point for the calculation, these operands may require that the number of fields be set to 1, then set Hx = 0 and Hy = 1, then finally choose Y Field 1 as the independent variable for a plot of the operand as a function of field. A similar trick works for getting plots as a function of wavelength.

Some individual operands, like DIFF and SUMM, have no meaning if selected from the "Operand" column because these operands are only defined when used as part of a larger merit function. If results from a calculation like DIFF are required, then the merit function should be selected from the Operand list and the individual DIFF's within the merit function selected for the "Line".

When selecting independent variables that are controlled by Multi-Configuration operands, note that the Multi-Configuration operand will override the changes made to the independent variable. For this reason Zemax automatically alters the corresponding data on the Multi-Configuration operand as well.

The escape key will terminate the analysis if the computation is taking too long.

Please note that since the Universal Plot analyses work on copies of the system data, macro commands that manipulate the user interface, such as CLOSEWINDOW, WINL(), and GETT() are not effective and should not be used in macros called by the Universal Plot using the ZPLM operand.

New Universal Plot 2D...

Displays as either a plot or as a text listing the value of any optimization operand as a function of two other parameters. See also "New Universal Plot 1D...".

Independent Variable X or Y Settings

Select either surface, system, multi-configuration, or NSC data as the independent variable.

Select the parameter to use as the independent variable. The available selections include radius, curvature, thickness, conic, parameter, and extra data values if surface parameters are chosen. For system parameters, the selections include the system aperture, field and wavelength data, apodization factor, temperature, and pressure. For configuration data, all multi-configuration operands are listed. For NSC data, object position and parameter data are listed.

The surface number for the independent variable parameter if the parameter is associated with a surface. This field is used for the configuration number if configuration data is selected. If NSC data is selected, this field is used for the object number.

Start/Stop Value The beginning and ending range of the independent variable.

# of Steps   The number of values between the start and stop values, inclusive, to compute the dependent variable function value. The minimum value is 2.

Dependent Variable Settings

Operand The optimization operand to use as the dependent variable function. The current system merit function is also included as "Merit". If the system merit function is selected, then either the entire merit function or any specific operand's value may be selected using the "Line" control below.

Line   The optimization operand # to use as the dependent variable function if "Merit" is selected for the Operand.

Int1/Int2 The Int1 and Int2 values for the selected operand. The dialog box will display the short name for the Int1 and Int2 values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Data1-Data4   The Data values for the selected operand. The dialog box will display the column heading name for these values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Display Settings

Show As   Choose surface plot, contour map, grey scale, or false color map as the display option.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String".

Min/Max Plot Scale   These controls define the min and max range of the Z-direction scale for the display. If the value is set to exactly zero, a default value will be selected. To define a value of zero, use a very small number such as 1.0E-20. If the defined plot scale values do not span the resulting data, the plot may clip, saturate, or otherwise distort the data display. These settings have no effect on the surface or contour plots.

Plot Title   Any text may be entered which will be used as the title for the plot or text listing.

Save As New Universal Plot   This will save the current settings in a Universal Plot 2D (UP2) file specified by the user. Any name which may be used as a valid file name may be entered. Although the "browse" window allows the file to be saved in any folder, Zemax will only recognize the file if it resides in the <data>\Miscellaneous folder (see "Folders").

Once the settings are saved, the file name will appear as a menu option under the Universal Plot menu option for quick regeneration of the plot.

Load From   This will load the settings from a previously saved Universal Plot 2D (UP2) file specified by the drop down list to the right of the button.

Discussion

This feature will either plot as a graph or create a text listing of the value of any optimization operand as a function of any two system, surface, or multi-configuration values.

For example, suppose a plot of the sagittal MTF at 30 lp/mm was needed as a function of the X and Y decenter of a lens group (which may be a useful diagnostic for tolerancing analysis). Since the MTFS operand computes the sagittal MTF, the Universal Plot 2D can generate such a graphic or text listing. See "Optimization operands" for a list of available optimization operands. The X and Y decenter of a lens group is defined by Parameters 1 and 2 on the relevant coordinate break surface, and Parameters 1 and 2 are both listed in the available surface group parameters.

Because of the number of different plots this feature can generate, there are no intelligent defaults for either the independent or dependent settings. These values need to be carefully provided in the settings dialog box. If the optimization operand cannot be computed, an error message will be displayed and the plot will not be generated.

Because many optimization operands accept Hx and Hy values to define the field point for the calculation, these operands may require that the number of fields be set to 1, then set Hx = 0 and Hy = 1, then finally choose Y Field 1 as the independent variable for a plot of the operand as a function of field. A similar trick works for getting plots as a function of wavelength.

Some individual operands, like DIFF and SUMM, have no meaning if selected from the "Operand" column because these operands are only defined when used as part of a larger merit function. If results from a calculation like DIFF are required, then the merit function should be selected from the Operand list and the individual DIFF's within the merit function selected for the "Line".

When selecting independent variables that are controlled by Multi-Configuration operands, note that the Multi-Configuration operand will override the changes made to the independent variable. For this reason Zemax automatically alters the corresponding data on the Multi-Configuration operand as well.

The escape key will terminate the analysis if the computation is taking too long.

Please note that since the Universal Plot analyses work on copies of the system data, macro commands that manipulate the user interface, such as CLOSEWINDOW, WINL(), and GETT() are not effective and should not be used in macros called by the Universal Plot using the ZPLM operand.

Applications Group

Stray Light
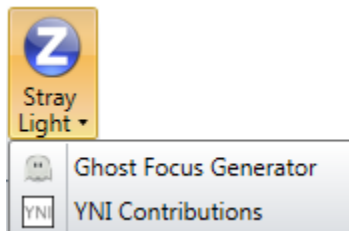
Ghost Focus Generator

Ghost focus analysis. For a description of ghosts, see "Ghost reflections".
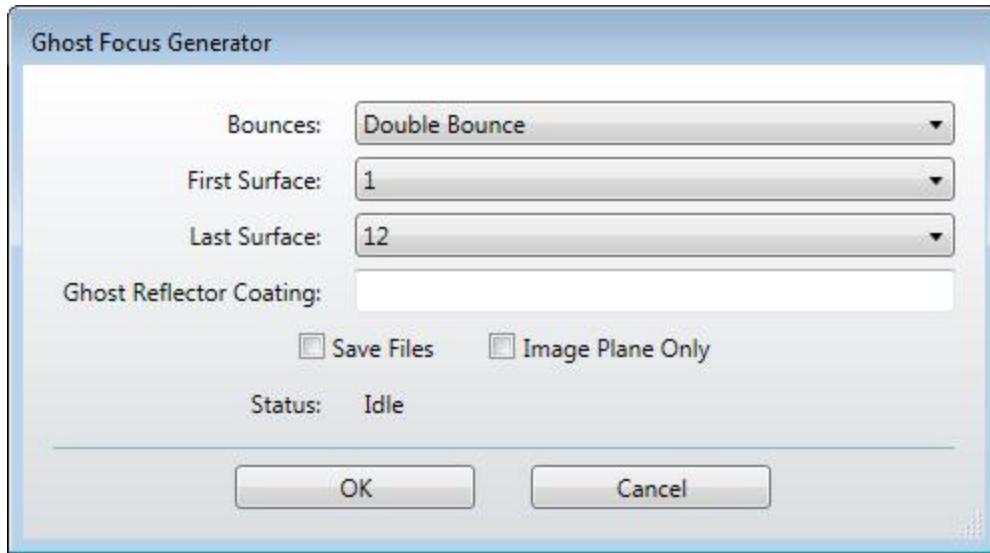
Bounces Select either single-bounce or double-bounce analysis.

First Surface The first surface to consider reflections from.

Last Surface The last surface to consider reflections from.

Save Files If checked, the files used to compute the ghost ray trace are saved to a file.

Image Surface Only If checked, only data for the image surface will be shown when computing double bounce ghosts.

Ghost Reflector Coating The name of the coating, if any, to apply to surfaces that will be changed from refractive to reflective to simulate ghosting. For example, if a 1% reflective coating is applied to all optical surfaces; then enter a coating name of "I.99". The result will be a surface that reflects 1% of the energy; and this will aid in accurate computation of total ghost energy using the polarization transmission analysis, if desired. This example uses an ideal coating, but real coatings may also be used as ghost coatings if the substrates are properly defined in the coating model. This feature is intended for computing detailed transmission data when looking at the generated and saved ghost files using polarization ray tracing and the transmission feature, see "Transmission".

Discussion

This feature may not work correctly for systems which contain coordinate breaks or some special surface types such as the non-sequential components surface. Only the current configuration is considered. Afocal mode is turned off for the purposes of this analysis.

This feature generates lens files which are derived from the current lens prescription data. The generated files are set up to reflect light at a given surface, rather than refract it. The portion of the optical system prior to the new reflective surface is then duplicated so the rays can be traced back through. The purpose of this analysis is to check whether or not rays reflected from any optical surface can form "ghost" images on other components or near the focal plane. These effects are significant in high power laser systems, where focused reflections can damage optics. Ghost images also reduce contrast. Both single and double reflections are supported.

An alternative to this type of ghost analysis is to use the non-sequential capabilities of Zemax. For details, see "NSC - OVERVIEW".

For each ghost system, the paraxial marginal ray height, paraxial ray F/#, and real ray RMS spot radius on axis are listed. Glass surfaces which may have an internal focus are also indicated. For the image surface, when doing double-bounce

ghost analysis, the paraxial marginal and chief ray heights, distance from the image surface to the ghost, and the EFL of the ghost system are also provided.

The lens files generated are stored in the file GHfffsss.ZMX, and these lens files can be opened for more analysis. The fff is the number of the first ghost surface, and sss is the number of the second ghost surface. For example GH007002.ZMX is a ghost focus file of the double bounce off surface 7 then 2. For single bounce ghosts, only the fff number is non-zero.

Dummy surfaces (those that do not define an index boundary) and mirror surfaces are ignored as candidate bounce surfaces. The only exception is the image surface, which is considered a potential first bounce surface even if there is no index change on that surface.

Not all ghost reflection configurations can be traced; there are occasionally problems with total internal reflection or rays completely missing surfaces. The GHfffsss.ZMX file may need to be opened and modified to analyze it in detail.

If the first reflected surface is before the stop, then the entrance pupil position is incorrectly calculated as well. This problem is easily overcome (for the purposes of analyzing the ghosts only) by using the following procedure before you begin the ghost focus generation:

1) Record the entrance pupil position and diameter.

2) Create a dummy surface at surface one.

3) Give this new dummy surface a thickness equal to the negative of the entrance pupil position value recorded.

4) Make the dummy surface the stop, and use an entrance pupil diameter equal to the entrance pupil diameter recorded.

5) For finite conjugates only, increase the object surface thickness by an amount equal to the thickness of the dummy surface.

These steps will give you a real entrance pupil in object space, and the rays will be correctly traced when reflected off surfaces before the stop. Ghost focus analysis can be very complex for moderately complicated systems, and care is required in interpreting the results of this analysis.

YNI Contributions

This feature lists for each surface the paraxial YNI value, which is proportional to the Narcissus contribution of that surface.
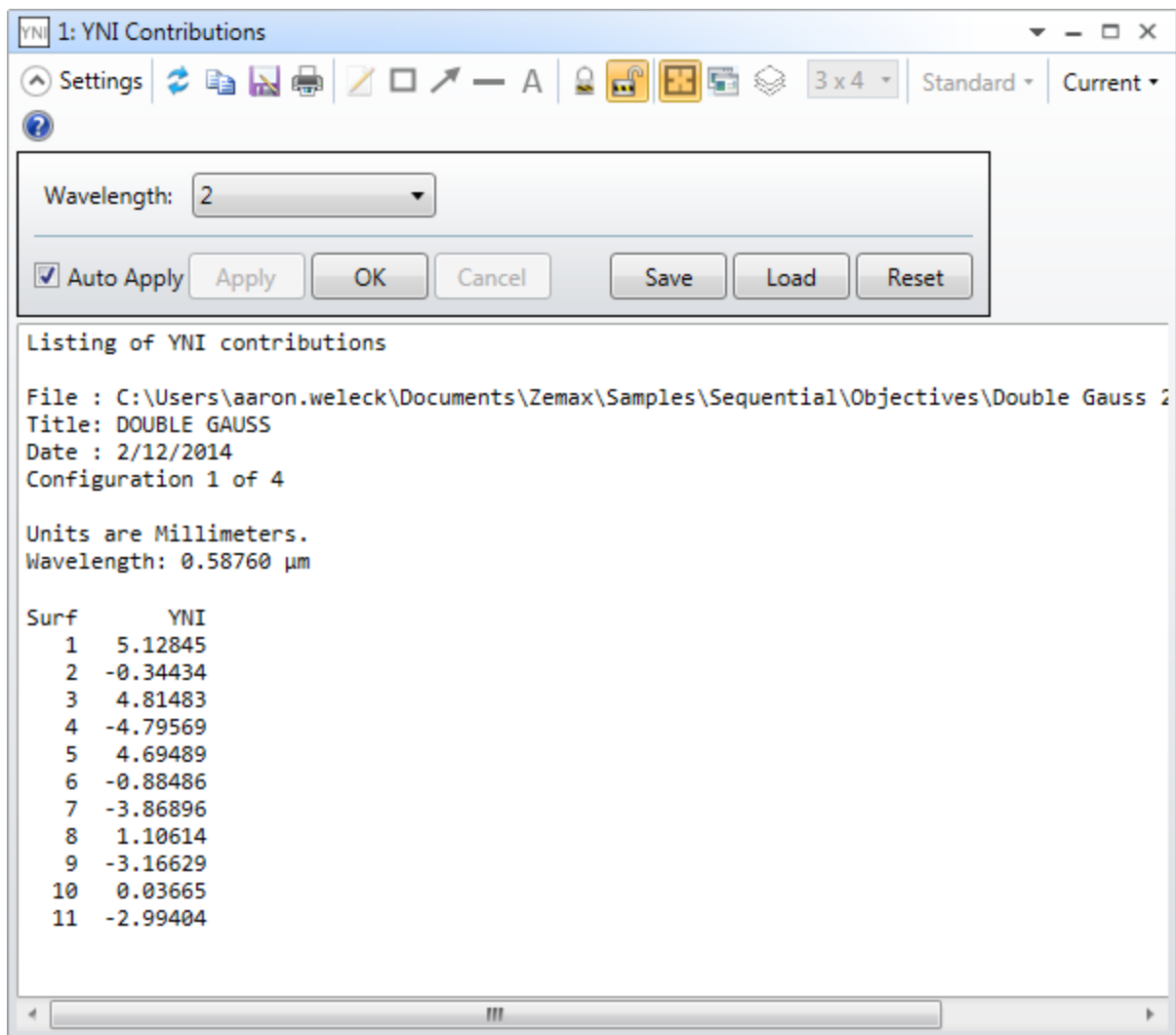
Wavelength   The wavelength number to use for the calculation.

Discussion See "Narcissus: reflections on retroreflections in thermal imaging systems", Applied Optics, Vol. 21, No. 18, p3393 (1982) for a discussion.

Biocular Analysis
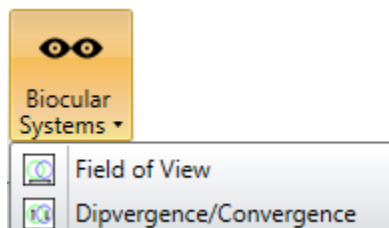
Analysis of biocular systems, where both eyes look through the same optical path to see a projected image.

Field of View

Displays the field of view for up to four configurations. See the discussion for important assumptions before using this feature.
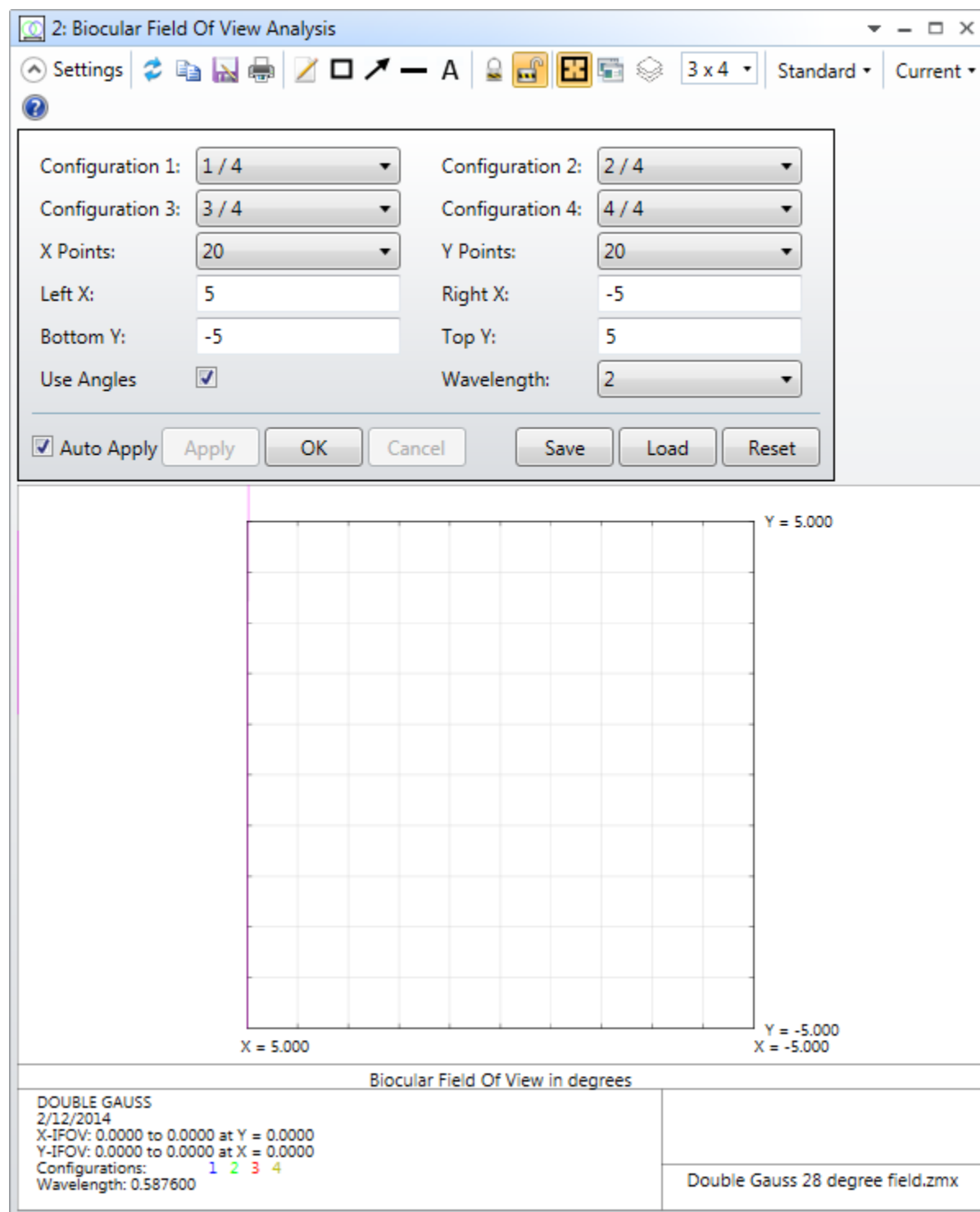
Config 1, 2, 3, 4 Selects which configurations to display the field of view data for.

X Points, Y Points Defines the number of points to use in each direction.

Left/Right X The left and right limits in field of view units.

Top/Bottom Y The upper and lower limits in field of view units.

Use Angles If checked, the field of view units are degrees, otherwise, the units are direction cosines.

Wavelength The wavelength number to be used in the calculation.

Discussion

This feature is intended to show the field of view for up to four configurations. Field of view in this context means the angle of rays emanating from the stop surface, not the object surface, which are unvignetted all the way to the image surface. The primary usage is the analysis of biocular systems, where both eyes look through the same optical path to see a projected image. This feature makes all of the following assumptions:

- The field of view is either angles in degrees or in direction cosines directly. The angles/cosines are measured along the chief ray from the stop position relative to the local Z axis at the stop surface. Surfaces prior to the stop are ignored.

- The image surface is assumed to be the location of the image the eyeball is looking at. Each selected configuration should represented a single (usually decentered) eyeball position.

- The eyeball decenter should be set up so that the X, Y coordinates on the image surface represent the same point on the image source in all configurations. For example, if the image source is a CRT, then the point with image coordinates (x = 1, y = 2) should correspond to the same physical location on the CRT in all configurations.

- All surfaces should have fixed apertures for the purposes of vignetting rays that are outside of the field of view.

The field of view is computed by tracing chief rays at various angles from each configuration. If the ray passes all apertures, then the point on the plot corresponding to that angle will be inside of the closed curve for that configuration. Each configuration will generate a closed curve representing the limits of the field of view for that configuration. The ray tracing is done iteratively to determine the exact angles at which the chief rays become vignetted.

If the field of view units are cosines, then the plot is linear in cosine space, with the Z direction cosine determined from the X and Y direction cosines. If the field of view units are angles, then the direction cosines of the chief rays are determined from the X and Y field angles using these formulas:

$$\tan(\theta_x) = \frac{l}{n}, \tan(\theta_y) = \frac{m}{n}, l^2 + m^2 + n^2 = 1$$

where l, m, and n are the X, Y, and Z direction cosines, respectively.

Dipvergence/Convergence

Displays the dipvergence and convergence for biocular analysis.
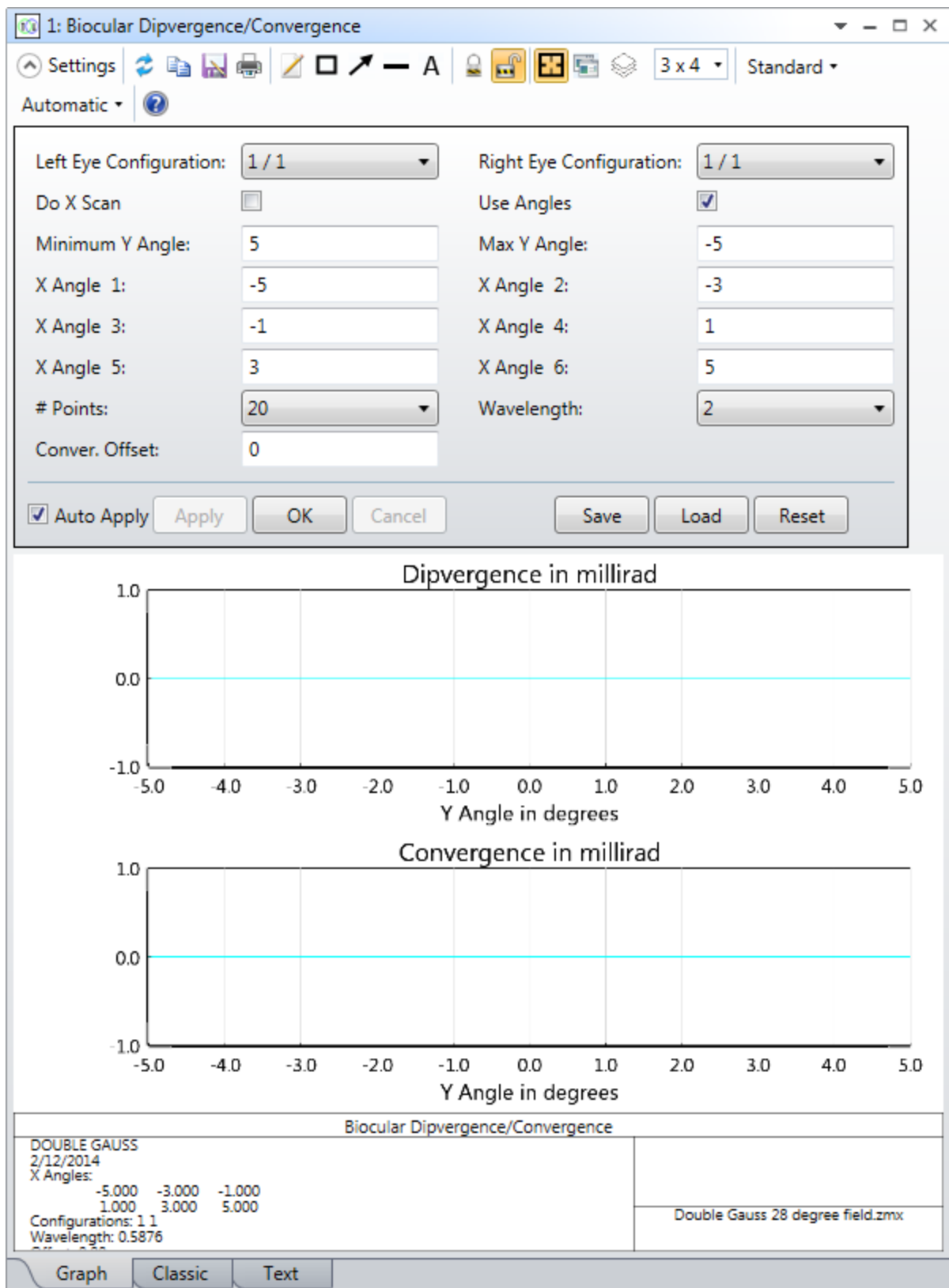
Left Eye Config Defines the configuration to use as the left reference eye.

Right Eye Config Defines the configuration to use as the right eye, whose dipvergence and convergence is determined relative to the left eye.

Do X Scan If checked, the plot scan is across the X direction field of view, otherwise, the scan is across the Y direction field of view.

Use Angles If checked, the field of view units are degrees, otherwise, the units are direction cosines.

Min/Max Angle / Cosine Sets the limits of the scan in X or Y, in either cosine space or in degrees. Note the limits do not need to be symmetric, and the min value need not be less than the max value.

X/Y Angle/Cosine 1-6 If "Do X Scan" is checked, then up to 6 Y angles/cosines offsets may be selected for display. If "Do X Scan" is unchecked, then up to 6 X angles/cosines offsets may be selected for display. To turn off one or more of the offset values, set the value to -99.

 # Points The number of points to use across the full scan.

Wavelength The wavelength number to be used in the calculation.

Convergence Offset This value is subtracted from the computed convergence value. This allows only the deviation from the desired value to be displayed.

Discussion

The assumptions made for this feature are the same as for "Field of View".

When two eyes are used to look through a biocular lens, there is usually a small angular difference between the direction the two eyes must look to see the same image point. The vertical (up/down) angle is called dipvergence. The horizontal left-right angle is called convergence if the eyes gaze toward a common point in front of the viewer's head, so the chief rays from the two eye positions would converge as they move toward the lens and away from the viewer's head. If the two chief rays diverge as the rays propagate toward the lens, as if to see a virtual image point behind the head, then the angle is called divergence. Convergence and divergence are really the same thing from a computational standpoint. Zemax uses the common convention that convergence is a positive value and divergence is a negative value; for this reason only the term convergence will be used in the subsequent discussion, with the understanding that if the convergence is negative then it is properly called divergence. Usually, convergence is more tolerable than divergence and the two aberrations may have different specifications. Dipvergence and convergence are both measured in milliradians, and typical limits are on the order of 1.0 milliradians for visual systems.

The computation proceeds for a given point in the field of view by tracing a reference ray in the left eye configuration. The same angle chief ray is then traced in the right eye configuration. In general, the right eye ray will not land on the exact same X, Y coordinates on the image surface as the left eye reference ray did. Zemax iterates the right eye trace until the chief ray is found that matches the left eye reference ray intercept coordinates. The resulting right eye chief ray will in general make some angle with respect to the left eye in both the vertical and horizontal directions, and it is these angles that are the dipvergence and convergence, respectively.

Note that both the left eye and the right eye chief rays must pass all surfaces without errors and without vignetting for the computation to be valid. If both rays do not trace, no data will be returned for that field of view. The field of view overlap between the configurations is very useful for setting appropriate min/max scan values, see the "Field of View" for determining the field of view overlap.

It is possible for the iteration required for the right eye to fail. This typically happens if either no solution exists or if the dipvergence/convergence is so large that algorithm becomes unstable. Failed iteration will usually show up as gaps in the plot.

PAL/Freeform
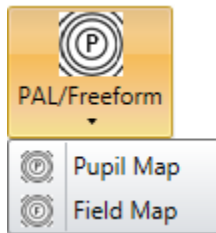
Power Field Map

This features computes optical power or effective focal length as a function of field position. A common application of this feature is analysis of the spherical and cylinder power of a Progressive Addition Lens (PAL). See also "Power Pupil Map".
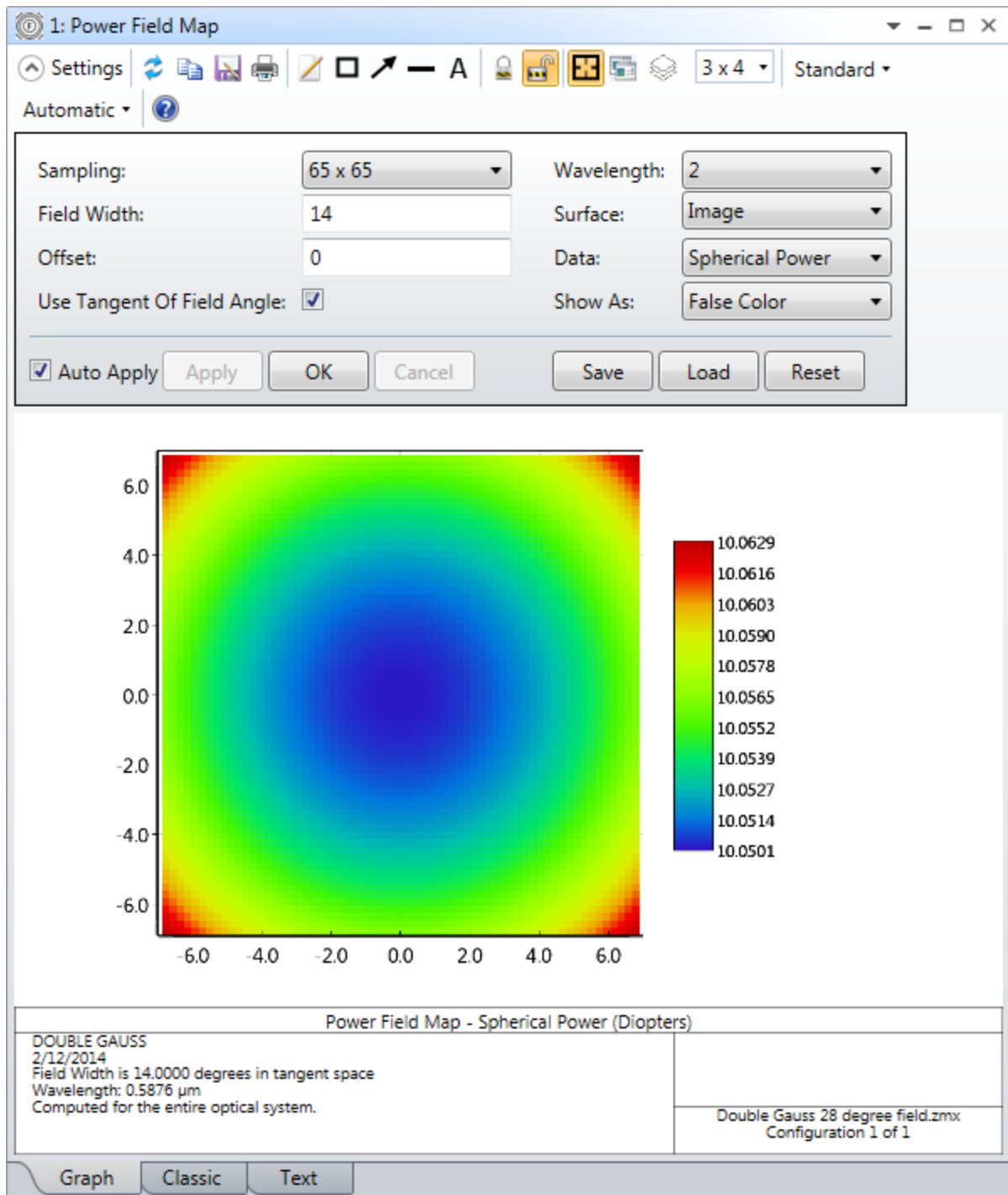
Sampling   The rectangular grid size to use for the analysis.

Field Width   The full width of the field in field units.

Offset   A user defined offset value that is added to the computed data.

Wavelength   The wavelength number to use for the analysis.

Surface The last surface with optical power. The power and EFL data is computed for the system as a whole up to and including refraction from this surface. See the discussion below.

Use Tangent of Field Angle If checked, and the field type is angle in degrees, the displayed spatial distribution of the data is proportional to the tangent of the angle rather than angle directly.

Data The data to be computed. The options are spherical, cylinder, maximum, minimum, tangential, sagittal, Y direction, X direction, and Astigmatic (X minus Y) optical power in diopters or effective focal length (EFL) in lens units.

Show As Choose surface plot, contour map, grey scale, or false color map as the display option. A scan across the X- or Y-field direction is also available.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String".

Discussion

This feature computes optical power or focal length as a function of field coordinate. The power or focal length is determined for the optical system as a whole up to and including refraction from any surface. The method used is to trace a ring of real rays around the chief ray at each point in the field. The ray data are used to determine the focal length for each field position. This focal length can then be used to compute the optical power in units of diopters (inverse meters). In the general case, the focal length is a function of orientation in the entrance pupil. By tracing a ring of rays, the average, maximum, and minimum optical power or focal length around the pupil can be determined. From this data, several types of optical power can be computed. Note that the definitions used by this feature may vary from other uses of these terms.

The spherical power is given by the average power over the entrance pupil. The maximum and minimum powers are the extreme values over the pupil. The cylinder power is defined as the difference between the maximum and minimum powers. Note using this definition the amount of cylinder power is always positive. For cylinder, the EFL value reported is the difference between the maximum and minimum focal lengths. For astigmatic EFL, the value reported is the difference between the X and Y direction focal lengths.

This feature can also compute the power along the tangential or sagittal directions. In this context, the tangential direction is along the plane that contains both the chief ray and the local z axis in object space. The sagittal direction is orthogonal to the tangential direction. Also available are the Y and X direction optical powers. The directions are defined by the local coordinate orientation on the entrance pupil. Astigmatic power is defined as the X direction power minus the Y direction power.

The surface setting should define the last surface of the portion of the lens being evaluated that has optical power. It is recommended that this be set manually to the correct value rather than use the default "Image" surface setting. The reason is that exact ray tracing is used to compute the optical power, and the most accurate data is the ray data immediately after refracting from the last powered surface. Zemax may not be able to correctly resolve some sign ambiguities if the rays propagate a long way past the lens.

The "offset" value is a user defined value added to all the computed data. This feature is useful for evaluating the deviation relative to some user defined reference value.

To optimize a lens for specific power characteristics, see the POWF optimization operand.

Power Pupil Map

This features computes optical power or effective focal length as a function of pupil position. See also "Power Field Map".
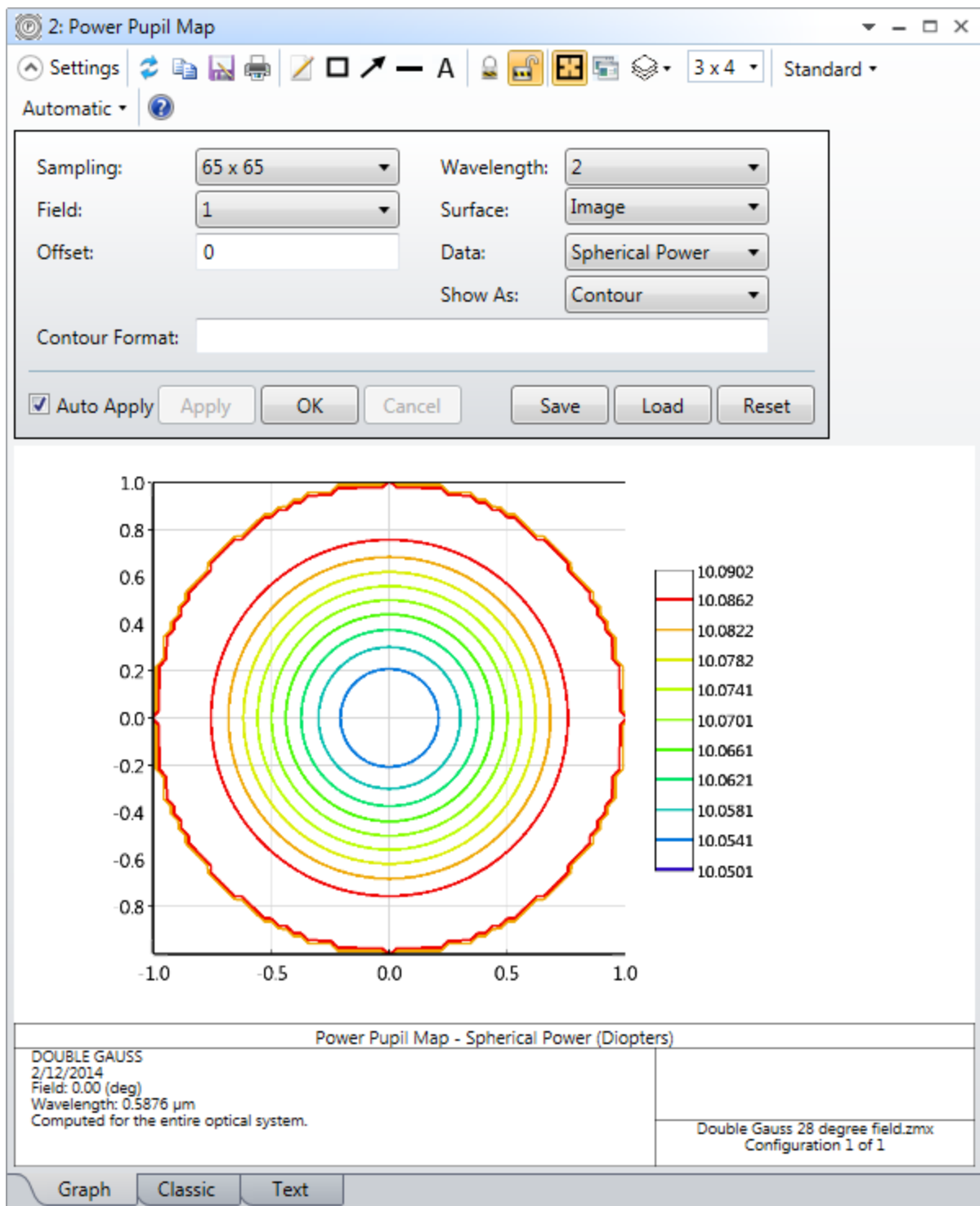
Sampling   The rectangular grid size to use for the analysis.

Field   The field number for which the calculation should be performed.

Offset   A user defined offset value that is added to the computed data.

Wavelength   The wavelength number to use for the analysis.

Surface The last surface with optical power. The power and EFL data is computed for the system as a whole up to and including refraction from this surface. See the discussion below.

Data The data to be computed. The options are spherical, cylinder, maximum, minimum, tangential, sagittal, Y direction, X direction, and Astigmatic (X minus Y) optical power in diopters or effective focal length (EFL) in lens units.

Show As Choose surface plot, contour map, grey scale, or false color map as the display option. A scan across the X- or Y-field direction is also available.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String".

Discussion

This feature computes optical power or focal length as a function of pupil coordinate. The power or focal length is determined for the optical system as a whole up to and including refraction from any surface. The method used is to trace a ring of real rays around a central reference ray at each point in the pupil. The ray data are used to determine the focal length for each pupil position. This focal length can then be used to compute the optical power in units of diopters (inverse meters). In the general case, the focal length is a function of orientation. By tracing a ring of rays, the average, maximum, and minimum optical power or focal length around the reference ray can be determined. From this data, several types of optical power can be computed. Note that the definitions used by this feature may vary from other uses of these terms.

The spherical power is given by the average power over the ring of rays. The maximum and minimum powers are the extreme values over the ring of rays. The cylinder power is defined as the difference between the maximum and minimum powers. Note using this definition the amount of cylinder power is always positive. For cylinder, the EFL value reported is the difference between the maximum and minimum focal lengths. For astigmatic EFL, the value reported is the difference between the X and Y direction focal lengths.

This feature can also compute the power along the tangential or sagittal directions. In this context, the tangential direction is along the plane that contains both the chief ray and the local z axis in object space. The sagittal direction is orthogonal to the tangential direction. Also available are the Y and X direction optical powers. The directions are defined by the local coordinate orientation on the entrance pupil. Astigmatic power is defined as the X direction power minus the Y direction power.

The surface setting should define the last surface of the portion of the lens being evaluated that has optical power. It is recommended that this be set manually to the correct value rather than use the default "Image" surface setting. The reason is that exact ray tracing is used to compute the optical power, and the most accurate data is the ray data immediately after refracting from the last powered surface. Zemax may not be able to correctly resolve some sign ambiguities if the rays propagate a long way past the lens.

The "offset" value is a user defined value added to all the computed data. This feature is useful for evaluating the deviation relative to some user defined reference value.

Graphics and Text Windows Operations

Graphics windows have a variety menu items that vary based on the type of analysis performed.

Here are screenshots of different toolbars on analysis windows:

Toolbar button functions
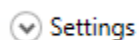
Settings	Opens settings window for the analysis.

Update	Refresh analysis window with the current values in editors
Copy	Copy data to the clipboard
Save As	Save the data as an image file
Print	Print the window

Annotate	See "Using the annotation feature" for details. The items under Annotate are:
Edit	Allows extended editing of annotations
Box	Draws a box on the graphic window
Arrow	Draws a line segment with a directional arrow at the end point.
Line	Draws a single line on the graphic window
Text	Prompts for and then draws text on the graphic window.

Lock/Unlock	If lock is selected, the window will be converted into a "static" window whose data cannot change. The locked window contents may be printed, copied to the clipboard, or saved in a file. The application for this feature is for comparing the results of different lens files. Once a window is locked, it cannot be updated, and so any new lens files which are subsequently loaded may be analyzed and compared to the locked window results. Once the window is locked, it cannot be updated until unlocked.

Active Cursor	The active cursor displays the value of the coordinates the cursor is currently pointing to in the title bar of the window when the cursor covers an "active" region of the displayed graphic. On most X vs. Y type graphs, the meaning of the displayed values is obvious. On some graphics, such as the 3D Layout, the displayed image is a projection of a 3D object on a 2D plane. The projection of the image

renders the coordinate data displayed by the active cursor less meaningful if the image has been rotated. Not all graphics support the active cursor. The active cursor is by default "off" but can be turned on and off by choosing this menu option. The active cursor can be set to automatically be on or off when a new graphic window is created on the Graphics menu of the Project Preferences settings found in the Setup Tab.

Clone This selection will open a new window whose settings and displayed data are initially identical to the current window. This feature is useful for creating a new window based upon the settings in the original window. The cloned window acts like any other window after it is created, so it may be updated or have its settings changed independent of the first window.
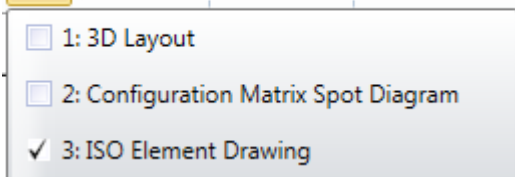
Overlay Provides a list of all open graphical windows; any of these may be selected for overlaying with the currently displayed data. The overlay feature is useful for comparing two similar graphs or layouts to detect small changes.
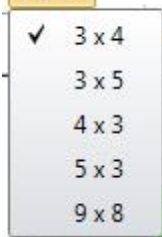
Aspect Ratio The aspect ratio may be selected to be 3 x 4 (height x width) which is the default, or 3 x 5, 4 x 3, or 5 x 3. The latter two are taller than they are wide. The default aspect ratio may be set on the Graphics menu of the Project Preferences settings found in the Setup Tab.

Reset Zoom Resets zoom to original setting.
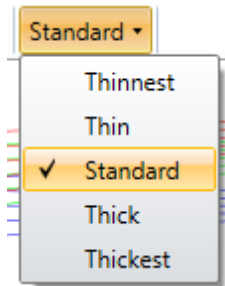
Line Thickness Set the thickness of lines of the analysis window
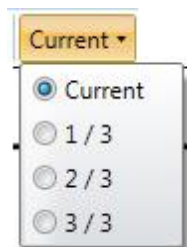
Configuration Set which configuration is shown.

Word Wrap Wrap text to window width. Also speeds up scrolling of very long files.

Help See more help on this feature
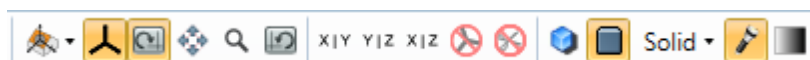
Shaded Model Viewer Controls
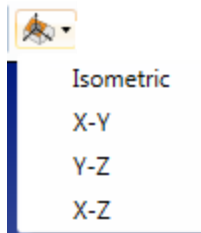
Camera View Adjust the camera orientation to view the object in an isometric or plane view.

Show Axis Show coordinate system axes orientation

Mouse control options

    Rotate   Use the left mouse-button and drag to freely rotate the object.

    Pan   Use the left mouse-button and drag to move the model left/right/up/down.

    Zoom   Use the left mouse-button and drag to select a rectangular zoom region.

    Roll   Use the left mouse-button and drag to roll the model.

Additional Camera Controls   Aside from using the Camera Control options within the drop down menu, the camera can also be manipulated manually using the mouse and/or keyboard.

    Middle mouse click + drag: Pan the camera

    Mouse-wheel: Zoom and Un-Zoom the camera

    Left & Right Arrow Keys: Rotate the entire scene clockwise or counter-clockwise

    Up & Down Arrow Keys: Rotate the Object about the X-Axis

    Page Up & Page Down Keys: Rotate the Object about the Y-Axis

X|Y Y|Z X|Z

Cutting Planes   Turn on a cutting plane along the X/Y, Y/Z, or X/Z directions.

Cutting axis manipulators open when cutting planes are used. These manipulators allow the cutting plane to be rotated and translated.
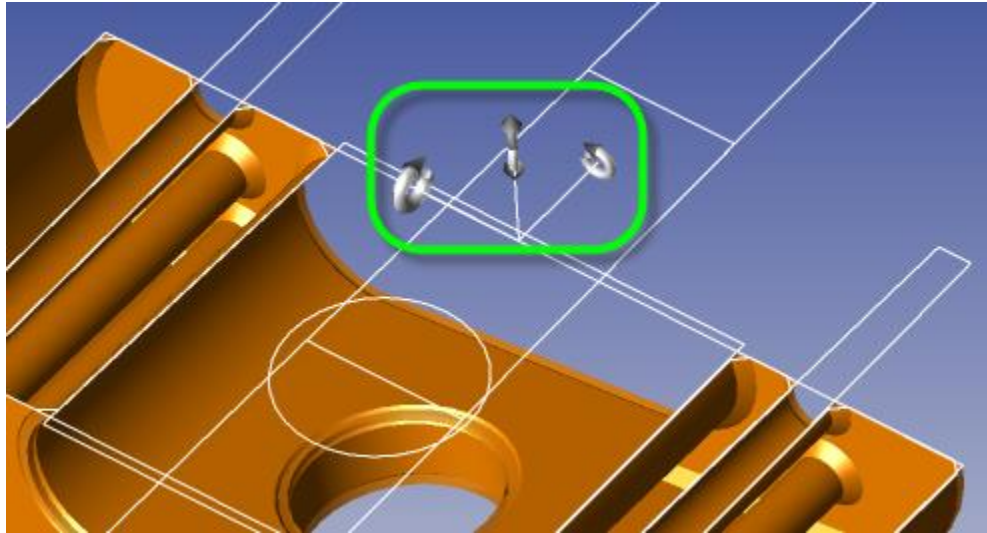
Hide Cutting Axes Hides cutting plane axis manipulators

Hide Cutting Plane   Hides the visualization of the cutting plane but keeps the cutting plane active.

Drop Shadow   Draw a shadow under the optical system based on the current lighting positions.

Shaded   Turn smooth shading on or off

Solid ▾

✓ Solid

    Hidden Lines
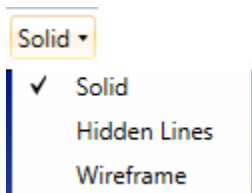
    Wireframe

Render Options   Change rendering options between Solid, showing Hidden Lines and Wireframe

Use Lighting   Toggles scene lighting on/off.

Use Simple Blending   Toggle between complex blending of transparent objects and simple screen-door blending.

OpticStudio's new graphics

Some of the analyses in OpticStudio have a new graphic display format, called LightningCharts. These charts have new features, including interactive axes and the ability to add data markers directly to a plot.

Classic Graphics

You can still view analysis charts as they were displayed in previous versions of Zemax. To do this, check the "Enable Classic View" setting in the Graphics section of the Project Preferences:
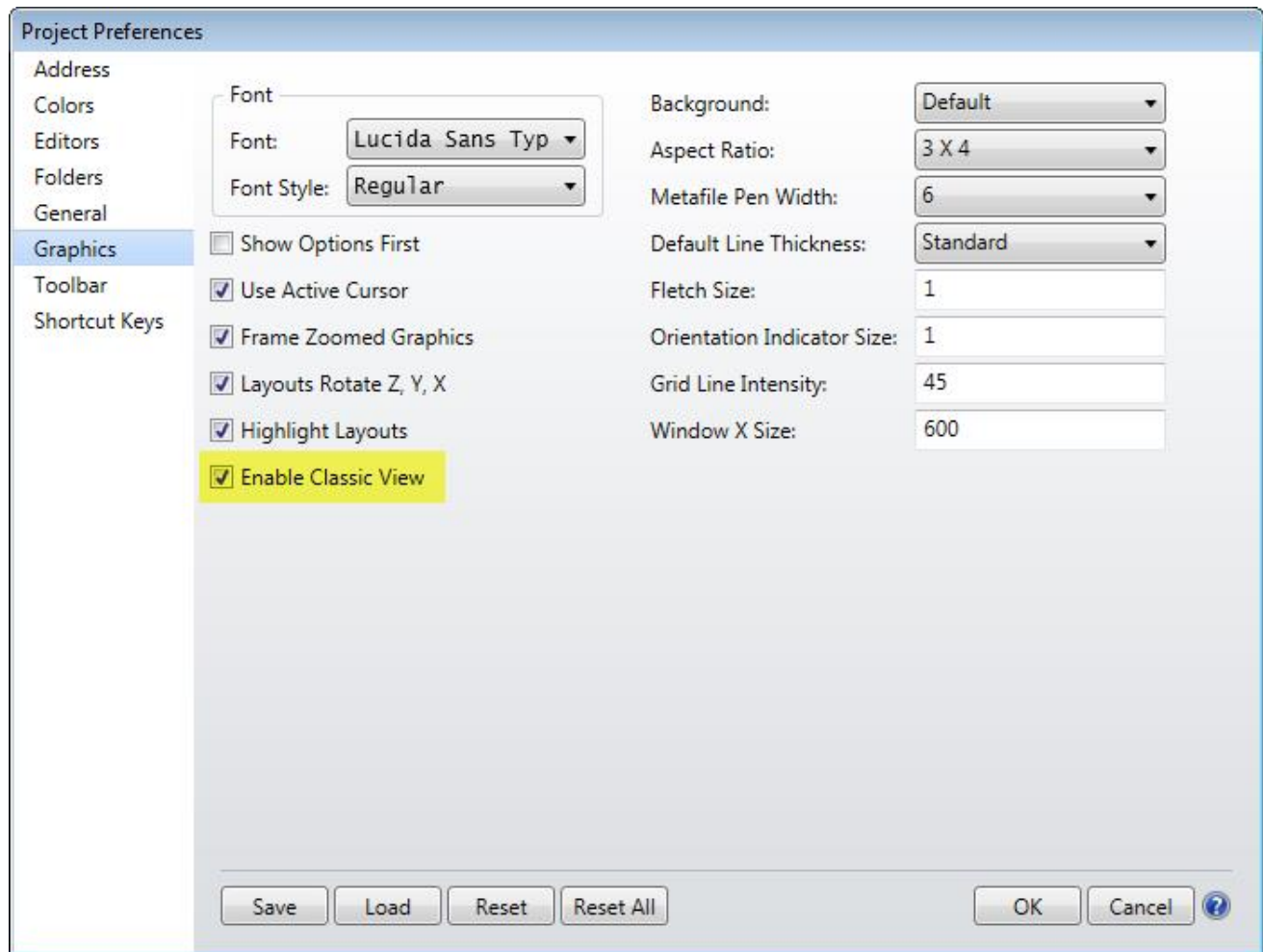
Checking "Enabling Classic View" enables a "Classic" tab at the bottom of LightningChart analyses, as shown in the following picture:
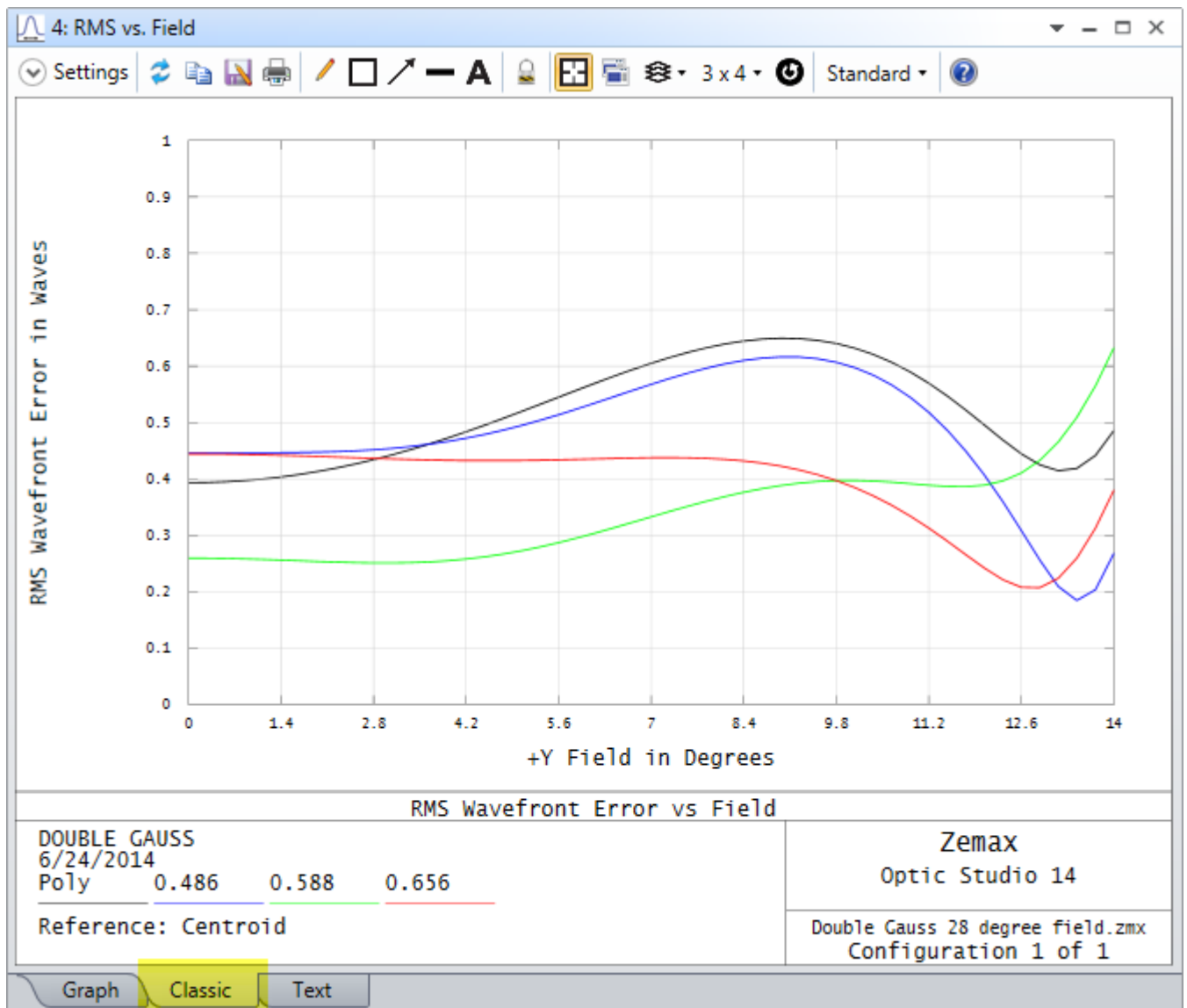
Right clicking on a Classic chart invokes the following menu. The tools listed in this right-click menu have the same functions as their matching buttons in the analysis toolbar:
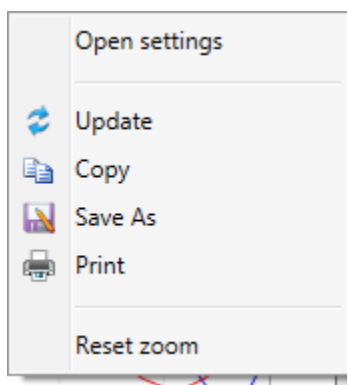
Note that the Classic tab is not available for analyses with graphics that have not been updated to the LightningChart format.

LightningCharts

For OpticStudio analyses with updated graphics, the Graph tab displays the data in a LightningChart.
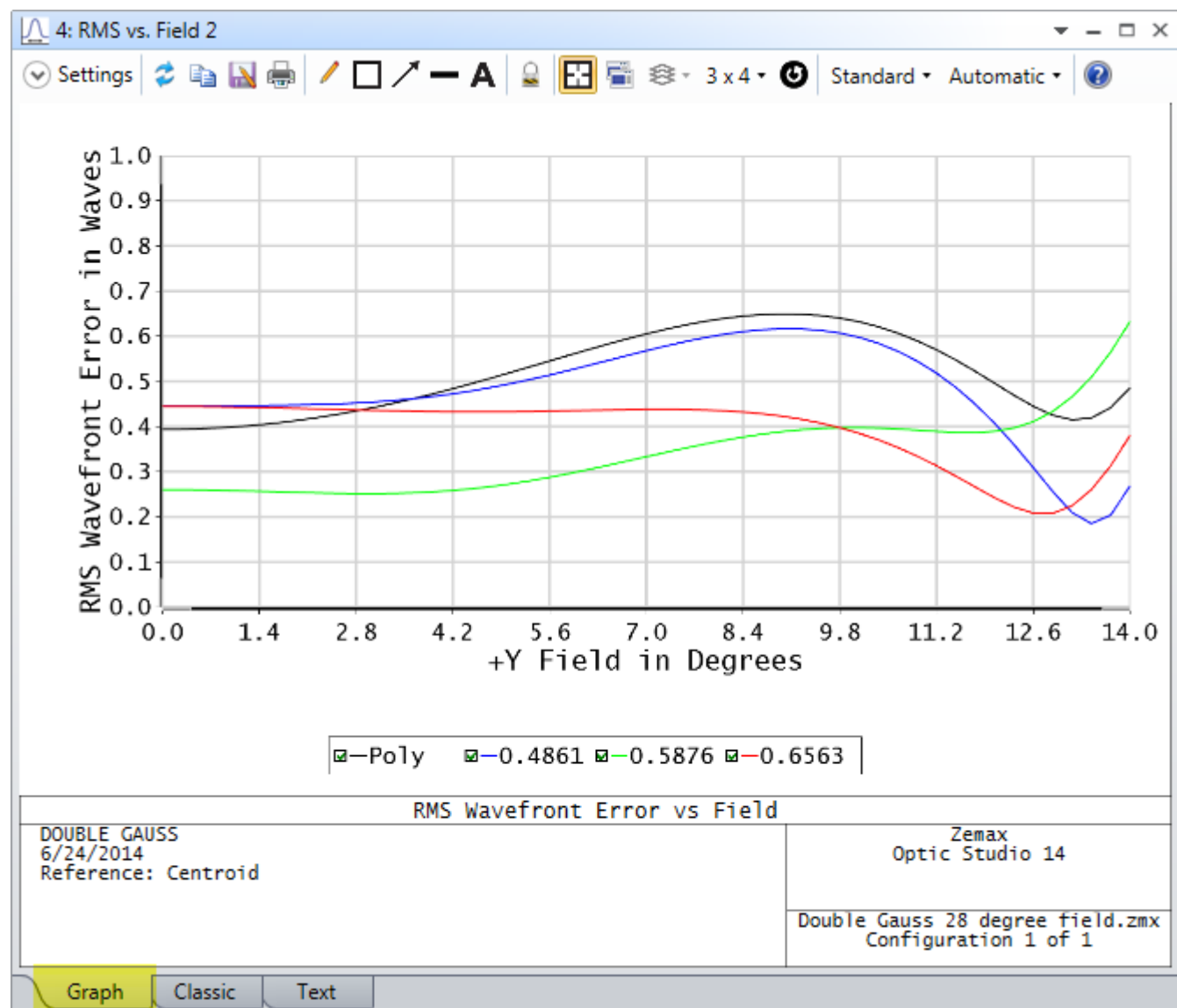
Toolbar differences with LightningCharts

Notice that there are two key differences between the Classic chart toolbar and the LightningChart toolbar:
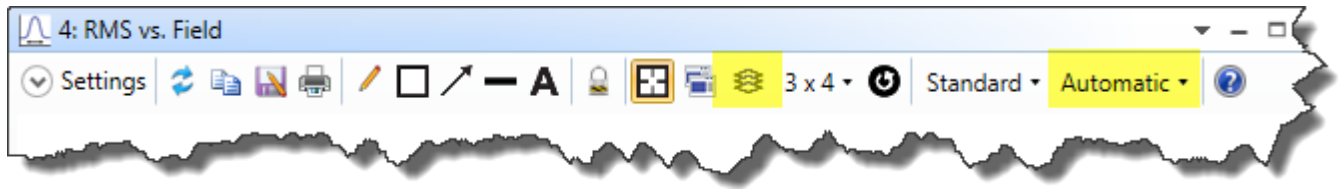
The overlay button is inactive in the LightningChart toolbar. This feature is only available for Classic charts.

The user can set the LightningChart resolution. This defines the resolution for the current graphic, and is useful for retaining readability when resizing a window. For small windows, user lower resolution.

Interactive features

LightningCharts also include interactive features.

Click and drag anywhere on an axis to move the range of displayed data:
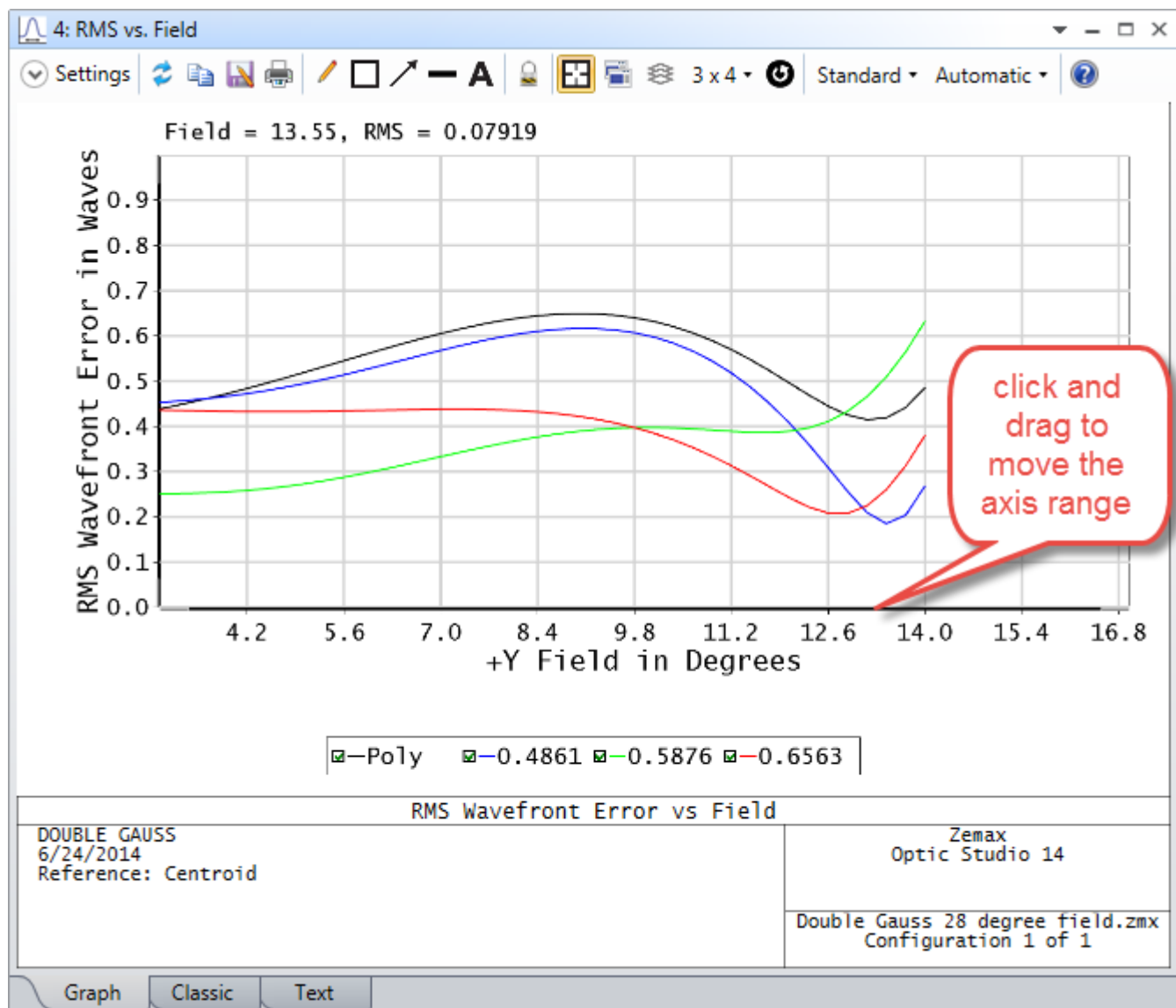
Click and drag the gray bars at the axis edges to increase or decrease the range of displayed data:
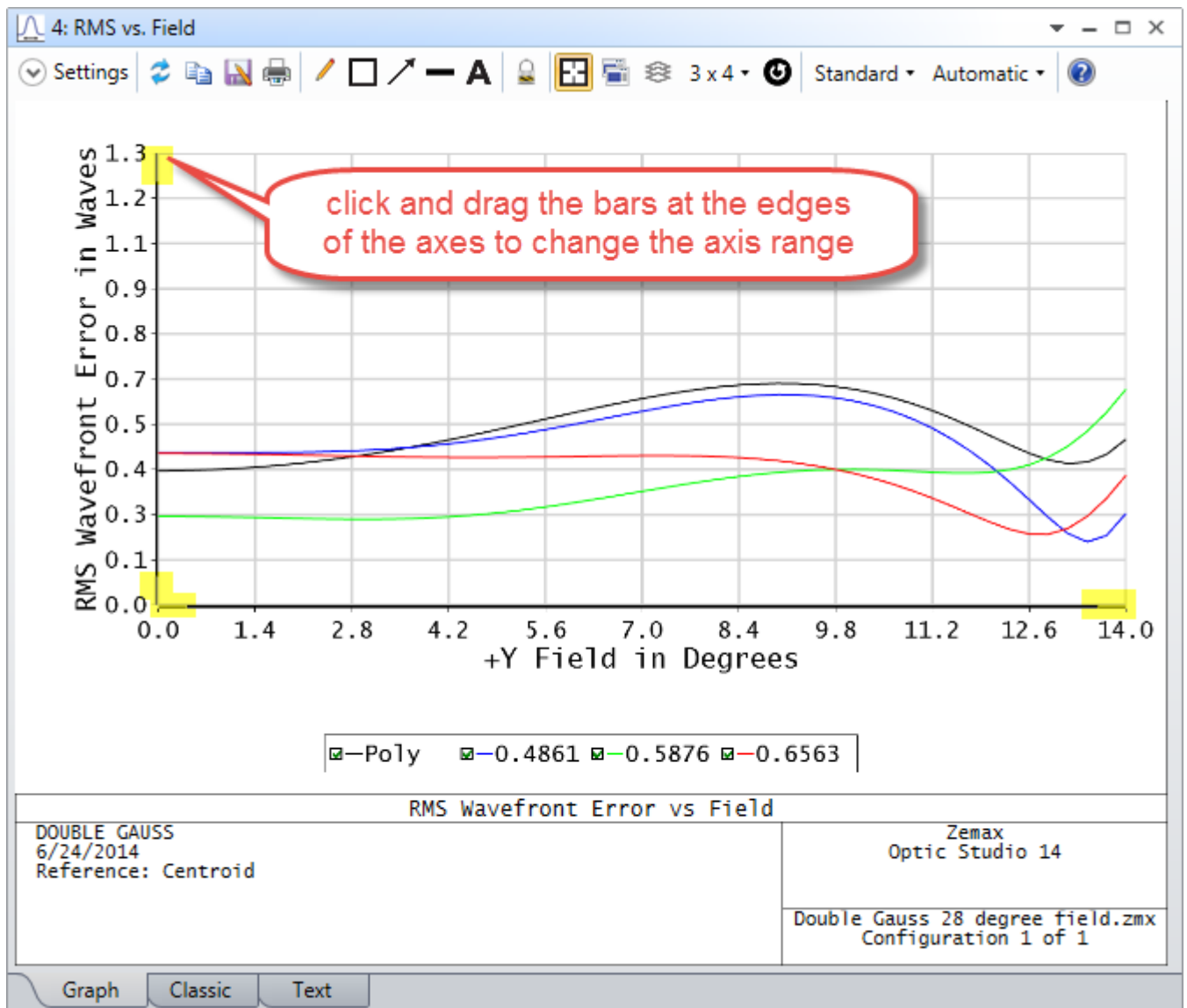
Right clicking on a Classic chart invokes the following menu.
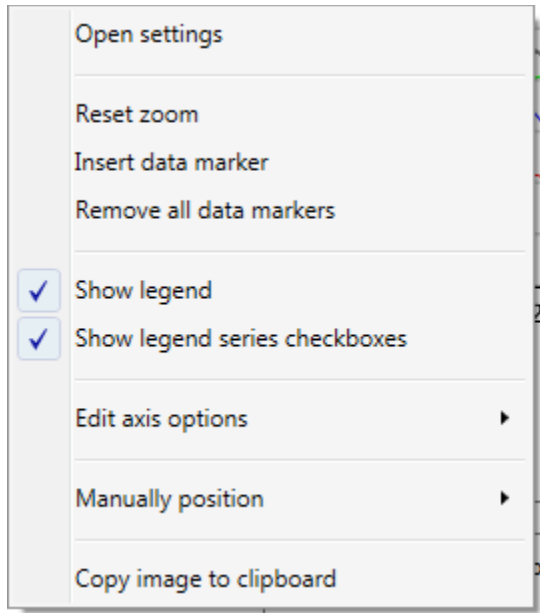
Data Markers

Insert data marker This option is enabled if the user right clicks on a plotted line, and it adds a marker with a data label to the chart.

Remove all data markers Removes all data markers from the chart.

Once a data marker is added to the chart, the user can click and drag to move the marker along the plotted line. The allowed data marker positions are directly tied to the calculated data points. There is no interpolation between calculated data points, so if the chart is zoomed in beyond the resolution of the these data points, the allowed data marker positions can become coarsely spaced.
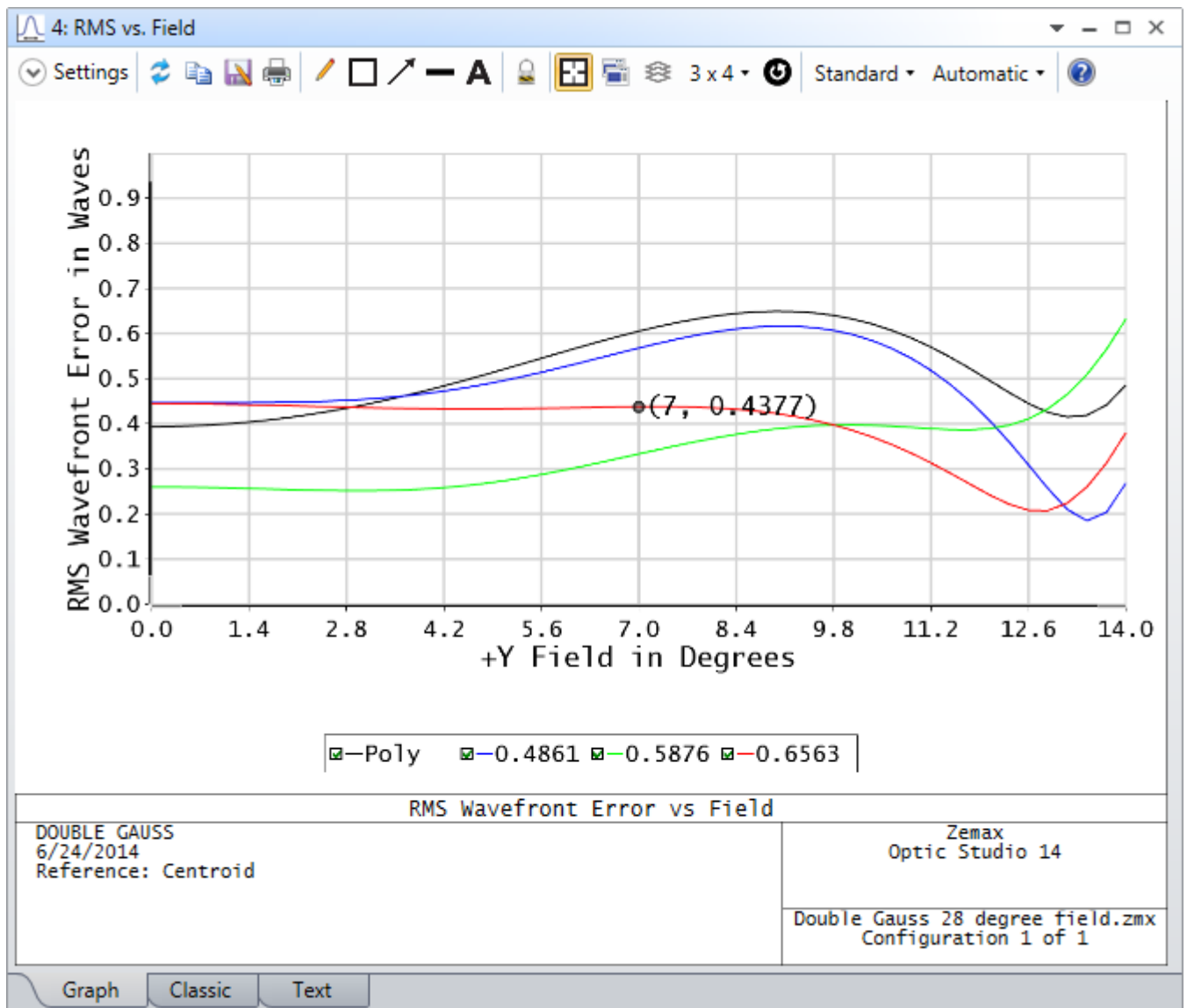
Legend options

Show legend Checked "on" shows the legend. Checked "off" hides the legend.

Show legend series checkboxes This feature allows the user to show and hide the different series by clicking a checkbox next to each series in the legend:
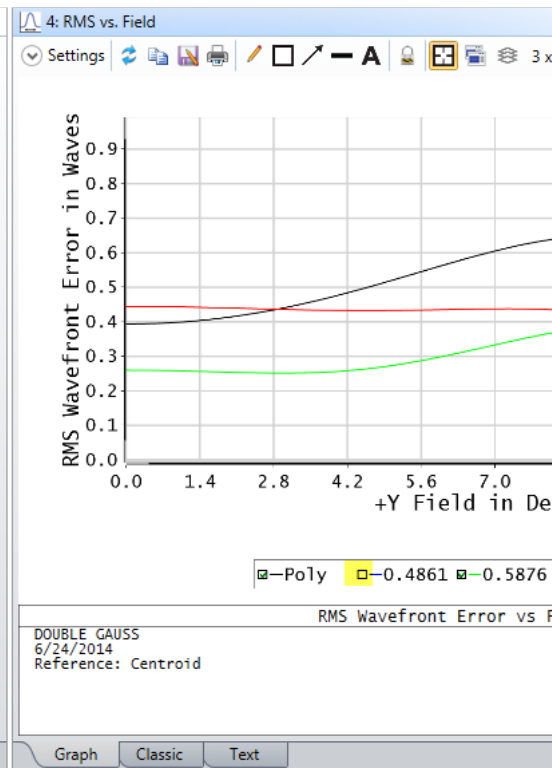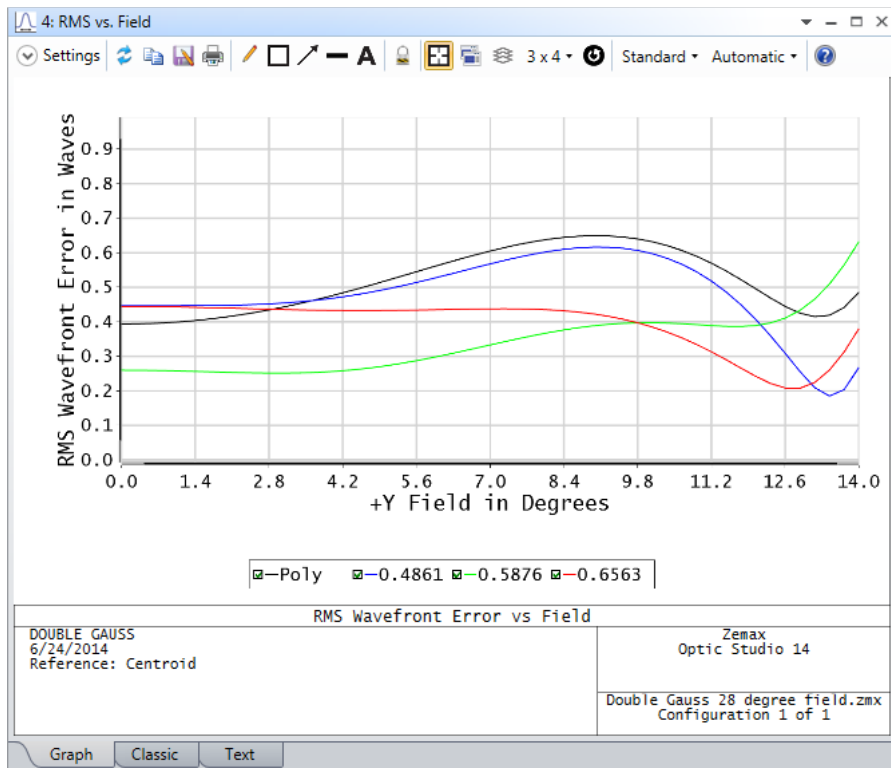
Manually position The user can manually position the legend by clicking and dragging.

Axis options

By default, the chart axes are auto-scaled and auto-titled, but the user can customize these settings by clicking Edit axis options… X or Y:
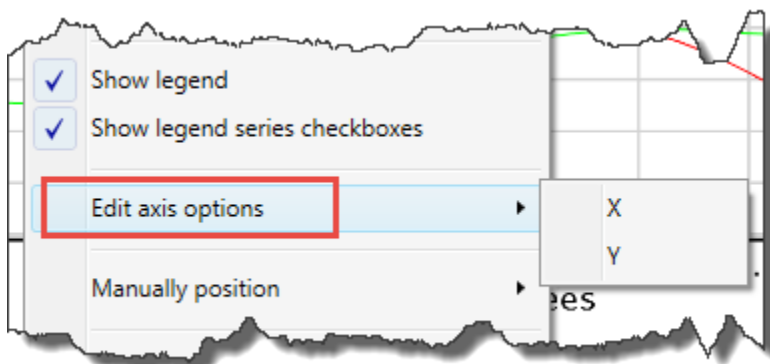
Uncheck the Auto option to enable the title and min/max values:
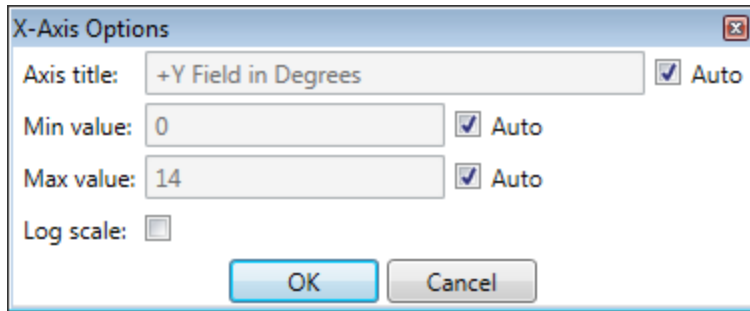
Using the Annotation Feature

There are several ways to annotate graphics windows with custom lines, arrows, boxes, and text notes. One way is to choose Window, Annotate from any graphic window menu bar, then select Line, Arrow, Text, or Box. To draw a line, choose the Line menu item, then click at the starting position of the line, holding down the left mouse button, then drag the crosshairs to the ending position of the line, then release the mouse button. A similar procedure will draw an arrow or box.

To add text to the window, select Window, Annotate, Text. A text entry dialog box will appear. Type in the desired text, then click "OK", then click on the desired location for the text on the window display area.

For more precise control over the exact locations of lines and text, as well as control over the text font, and the ability to add more complex annotations, choose Annotate, Edit from the graphic menu. This will invoke the annotation editor, which consists of a text editor and several buttons. There is also a single checkbox for enabling or disabling annotations for the graph.

The text edit field is used to define the annotations to be applied to the graph. To insert a new line, use the keyboard command Ctrl-Enter.

There are several supported commands, each with a specific syntax:

```
TEXT "string" x y angle fontx fonty
```

The TEXT command will write any text within the double quotes at the location specified by x and y, at an angle in degrees given by angle, using a fixed font whose width and height are given by fontx and fonty. The coordinates are in normalized units, where the left edge of the graph has a coordinate of x = 0.0, the right edge x = 100.0, the bottom edge is y = 0.0, and the top edge is y = 100.0. The origin is the lower left hand corner. The "angle" value is in degrees. The fontx and fonty units are in arbitrary units. The angle, fontx, and fonty values may all be left undefined, and then default values will be used.

```
LINE x1 y1 x2 y2
```

The LINE command draws a straight line from x1, y1 to x2, y2. The units and coordinate system are as described for x and y in the description of the TEXT command.

```
ARROW x1 y1 x2 y2 SIZE
```

The ARROW command draws a one-headed arrow pointing from x1, y1 to x2, y2. The units and coordinate system are as described for x and y in the description of the TEXT command. If Size is 1.0 or is omitted, the arrow head is drawn at the default size. To scale the default arrow head size, use any other floating point value for size. For example, a size value of 2.0 will make the head twice as large as the default, a value of 0.5 will make it half the default size, etc.

```
BOX x1 y1 x2 y2
```

The BOX command draws a box with opposite corners from x1, y1 to x2, y2. The units and coordinate system are as described for x and y in the description of the TEXT command.

```
ELLIPSE x y rx ry
```

ELLIPSE draws an ellipse centered at x and y with x half width of rx and half height of ry. If ry is the same as rx or if ry is omitted, a circle of radius rx results.

There are also several buttons on the annotation dialog box:

OK: Accepts the annotations as displayed and exits. Cancel: Reverts back to the last annotations and exits.

Save: Opens a "Save As" type box where the file name may be specified. The annotations are saved in the user named file.

Load: Opens a "Load: type box where the file name to load may be selected. The loaded file contains the annotations to use.

Reset: Clears the edit buffer.

Using the Windows clipboard

One of the most useful features in Windows is the clipboard. The clipboard is "holding area" for graphics and text. The advantage to using the clipboard is that virtually all Windows programs can either import or export to the clipboard.

Since Zemax is primarily used to generate graphical and text data, Zemax only supports exporting to the clipboard. Once the desired data has been copied to the clipboard, it is easy for another application, such as a word processor, graphics editor, or desktop publishing system to retrieve the data.

To get Zemax graphics and text into the clipboard, choose the graphic or text window desired, then select the copy button from the Toolbar. Nothing will appear to happen (the data transfer is extremely fast), however the data will now be available to other applications.

To now get the clipboard data into a document processing application, run that application, and choose Paste, usually from the Edit menu of that application. See the documentation for that application for details.

Another way of getting Zemax graphics into other applications is to perform a screen capture which creates a bitmap image of either the entire display or any single window. To capture the entire screen as a bitmap image, use the Print Screen button on the keyboard (and this button varies by keyboard). To capture a single window, select that window and press Alt-Print Screen. Once the image has been captured, the image can usually be pasted into another application using Ctrl- V or Edit, Paste, depending upon the application.

Using pan and zoom

Any graphic window may be panned (scrolled left, right, up and down) or zoomed (magnified). To activate the pan and zoom, choose any Zemax graphic window, then click the left mouse button and hold the button down for 1/2 second anywhere in the window. The cursor will change from an arrow to a cross. Now drag the mouse down and right to define a rectangle of the desired size which covers the area to zoom in on. Now, let go of the left mouse button. The selected area will be magnified to fill the window, while maintaining the aspect ratio of the plot.

Zooming may also be accomplished via scrolling the mouse wheel

Panning is accomplished via a right click and hold for analysis windows, or scroll wheel click and hold for shaded models and NSC viewers, then moving the mouse. Panning is only allowed after a graphic has been zoomed.

To restore the graphic to the original size, right click the analysis window and select "reset zoom".

Aborting long computations

Certain Zemax tools may require relatively long computation times. For example, the local optimization, global optimization, and tolerancing tools may run from several seconds to many days. To terminate the execution of these tools, there is a "Terminate" button displayed which can be clicked. After the terminate button is clicked, Zemax gracefully exits the computation and returns control to the main program. Usually, the results of the computation are not available, and are not displayed.

Some analysis features, such as MTF and the image analysis feature, will run for long periods of time in some circumstances. For example, large MTF grids or large image analysis ray densities require longer computation times. However, analysis features do not display a status box or terminate button, because analysis features display their output directly in a window. For this reason, the keyboard command "Escape" is used to terminate lengthy analysis computations. There is no mouse equivalent for this feature; only the Escape key is used.

The Escape key will terminate MTF, PSF, encircled energy, and other diffraction computations. If the Escape key is pressed, control will return to the main program (it may take 1 or 2 seconds) and the data displayed in the window will be invalid. For the image analysis feature, the escape key will terminate the tracing of new rays, however, rays that have already been traced will be displayed, and data from those rays is accurate, if incomplete.

The Analyze Tab (Non-Sequential UI Mode)
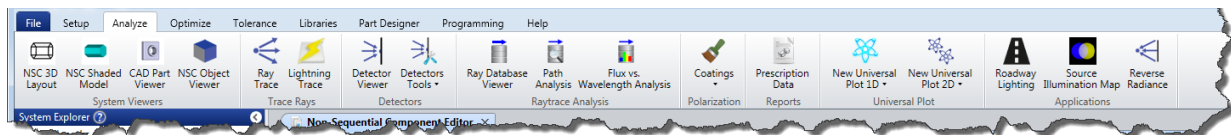
System Viewers Group
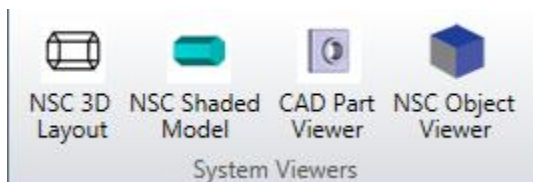
NSC 3D Layout

This feature is only available in the Professional and Premium editions of Zemax.

Draws 3D layout plots of the sources and objects in a single NSC group.
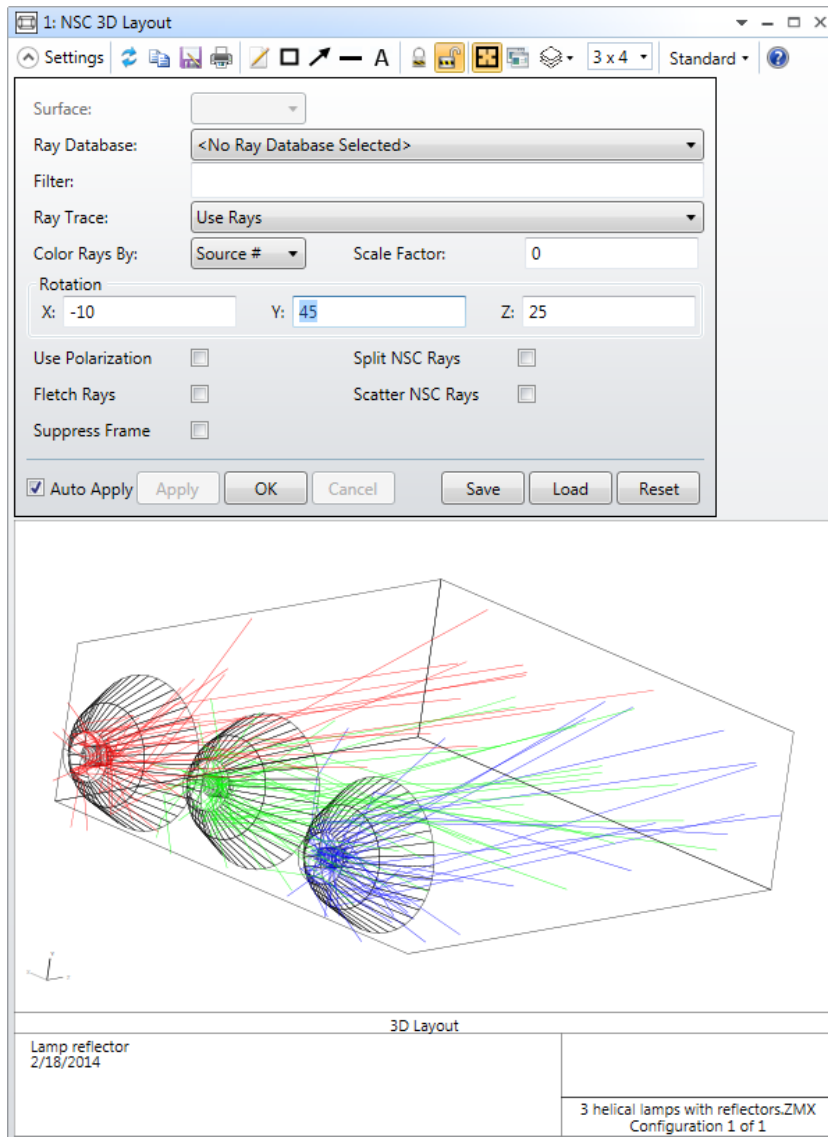
Fletch Rays If checked, small arrows are drawn on each ray to indicate the direction of propagation.

Split Rays If checked, rays from NSC sources will be statistically split at ray-surface intercepts. Rays entering from the entry port are not affected by this setting.

Scatter Rays If checked, rays from NSC sources will be statistically scattered at ray-surface intercepts. Rays entering from the entry port are not affected by this setting.

Use Polarization If checked, polarization is considered. See "Polarization" for information on defining the polarization state and how polarization is used by analysis features.

Suppress Frame Suppresses drawing of the frame on the bottom of the window, which leaves more room for the layout plot itself. No scale bar, address block, or other data will be displayed.

Configuration Select "All" to show all configurations, "Current" to show just the active configuration, or select any combination of other configurations.

Color Rays By Select "Source #" to use color to distinguish rays traced from each source, unless a source has a user defined color assigned to that source, "Wave #" to distinguish between each wavelength, "Config #" to distinguish

between configurations, "Wavelength" to approximate the color of wavelengths in the visible spectrum, or "Segment #" to distinguish between segment level number (number of object intersections since the ray left the source point).

Surface   Selects the NSC surface.

Scale Factor   If the scale factor is set to zero, then "Fill Frame" will be selected, which will scale the range of surfaces drawn to fill the graphic page. If a numeric value is entered, then the plot will be drawn in "real" scale, times the scale factor. For example, a scale factor of 1.0 will plot the lens actual size on the printer (not the display). A factor of 0.5 will plot the lens at half scale.

Rotation About X   The angle in degrees by which the lens appears to be rotated about the X axis.

Rotation About Y   The angle in degrees by which the lens appears to be rotated about the Y axis.

Rotation About Z   The angle in degrees by which the lens appears to be rotated about the Z axis.

Offset X, Y, Z   The X, Y, and Z direction offset between configurations in lens units. Only has an affect on the drawing if "All" configurations are being drawn.

Ray Trace   If the default option of "Use Rays" is selected, conventional ray tracing is used to draw rays on the plot, with the number of rays being determined by the "# Layout Rays" parameter values for all sources defined in the NSC group.

If one of the "Use LightningTrace" options is selected, rays from the LightingTrace mesh are drawn on the plot (see "The LightningTrace Control"). The mesh rays that are drawn correspond to a Ray Sampling input of "4X" and an Edge Sampling input of "Low (1X)" (the Ray Sampling may be automatically reduced by Zemax to "Low (1X)" if the total number of rays in the mesh exceeds the total number of rays that can be rendered on the plot).

For "Use Lightning Trace - True Color", the color of each ray in the plot is determined from the Tristimulus values for the ray (the ray color will be black if the Tristimulus values are zero).

For "Use Lightning Trace - Average Wavelength", the color of each ray in the plot is based on the source from which the ray is launched; the color corresponds to the average wavelength of all rays from that source.

Filter   If blank, all rays are drawn. Otherwise, only rays which meet the criteria defined by the filter string will be drawn. See "The filter string" for information about the filter string syntax.

The name of the filter string being used is shown in the bottom right corner of the layout plot. If the filter being used is a path filter, the path number and associated filter may be incremented by pressing the space bar. Holding the control key while pressing the space bar decrements the path filter, and also holding the shift key will increment or decerement the path number by 10. For more information, see "Path Analysis" .

Ray Database   If "none" is selected, new rays will be traced and displayed. If a ZRD data base file is selected, then rays contained within the database will be displayed. In either case the filter, if any, is applied. Generally, reading rays from a large database is faster than retracing them. The other advantage to using a database of rays is that the rays are always the same, until the database is replaced. Zemax cannot tell if the selected database is for rays from the current lens being displayed; so care should be taken when selecting the ZRD file that it corresponds to the lens data being displayed. For more information on the ZRD file, see "Ray database (ZRD) files" .

Discussion

The settings in the above table are very much like their counterparts on the 3D Layout feature. However, this feature only draws objects in a single NSC group, and only traces and draws rays from sources defined within the group.

To make surfaces partially transparent, see "Surface properties type tab" .

NSC Shaded Model

This feature is only available in the Professional and Premium editions of Zemax.

Draws a shaded solid model representation of the components in a non-sequential group.
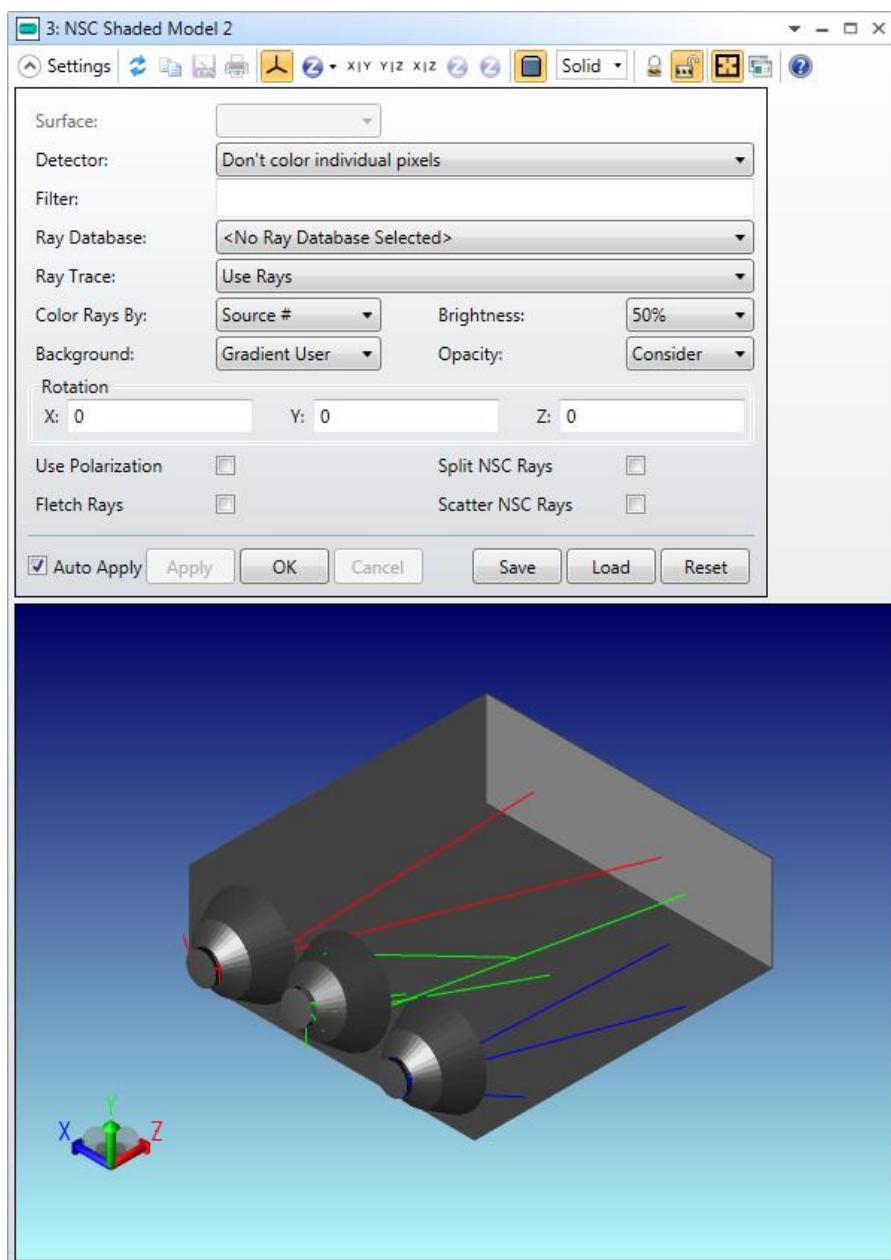
The options are very similar to those available for the NSC 3D Layout and Shaded Model features. One additional feature is the option to color detector objects by either the energy incident on the detector in the last analysis or by just the rays traced in the layout view. The detector shows false color or black and white displays, using either coherent intensity or any of the other options supported by the detector viewer. See "Detectors" for details.

CAD Part Viewer

This feature is only available in the Premium edition of Zemax.

Draws a shaded solid model representation of a part or assembly as defined in a CAD format file

File Name   Selects the CAD part file to draw.

Brightness   This control varies the brightness of the image.

Opacity The options are to ignore, which draws all surfaces and objects as opaque, or all 50%, which draws partially transparent objects.

Background   Selects the background color.

Rotation About X   The angle in degrees by which the lens appears to be rotated about the X axis.

Rotation About Y   The angle in degrees by which the lens appears to be rotated about the Y axis.

Rotation About Z   The angle in degrees by which the lens appears to be rotated about the Z axis.

Discussion

The object may be rotated with the mouse or the cursor keys to change the viewing angle. When selecting a SolidWorks (*.sldprt, *.sldasm), Autodesk Inventor (*.ipt, *.iam), or Creo Parametric (*.prt, *.asm) file, the file is automatically converted to either a STEP or an IGES file for viewing.

NSC Object Viewer

This feature is only available in the Professional and Premium editions of Zemax.

Draws a shaded solid model representation of a single NSC object.
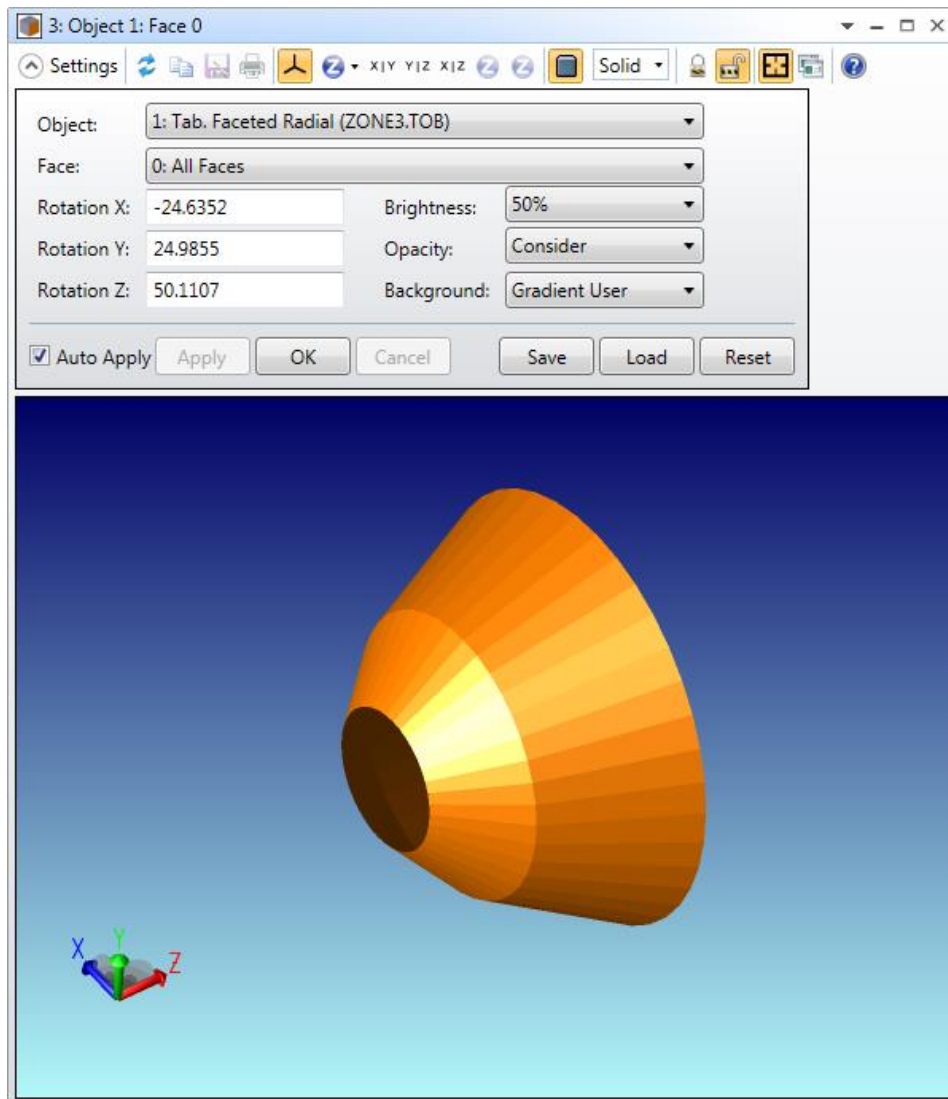
Object Selects the object to draw.

Face/Surface Selects the face (or surface) of the object to highlight. All portions of the object that belong to the selected face (or surface) will be shaded for identification. The highlighted face (or surface) can be incremented or decremented while viewing the object using the keyboard command Shift+Page Up/Down, respectively.

Note surface in this context refers to a single surface entity in the object, and not the

NSC Surface that contains the entire Non-Sequential Component group of objects.

Highlight Select either Face or Surface. If Face is selected, the Face/Surface item above selects the face to highlight, otherwise the item selects the surface entity.

Surface Selects the NSC surface.

Brightness   This control varies the brightness of the image.

Opacity The options are to ignore, which ignores the opacity values and draws all surfaces and objects as opaque; or consider, which draws partially transparent surfaces and objects. For information on setting the opacity of objects see "The object properties dialog box" .

Background   Selects the background color.

Rotation About X   The angle in degrees by which the lens appears to be rotated about the X axis.

Rotation About Y   The angle in degrees by which the lens appears to be rotated about the Y axis.

Rotation About Z   The angle in degrees by which the lens appears to be rotated about the Z axis.

Trace Rays Group

Ray Trace

The Ray Trace Control is used to launch and trace rays for analysis using defined sources, objects, and detectors. The controls on the dialog box are described below.
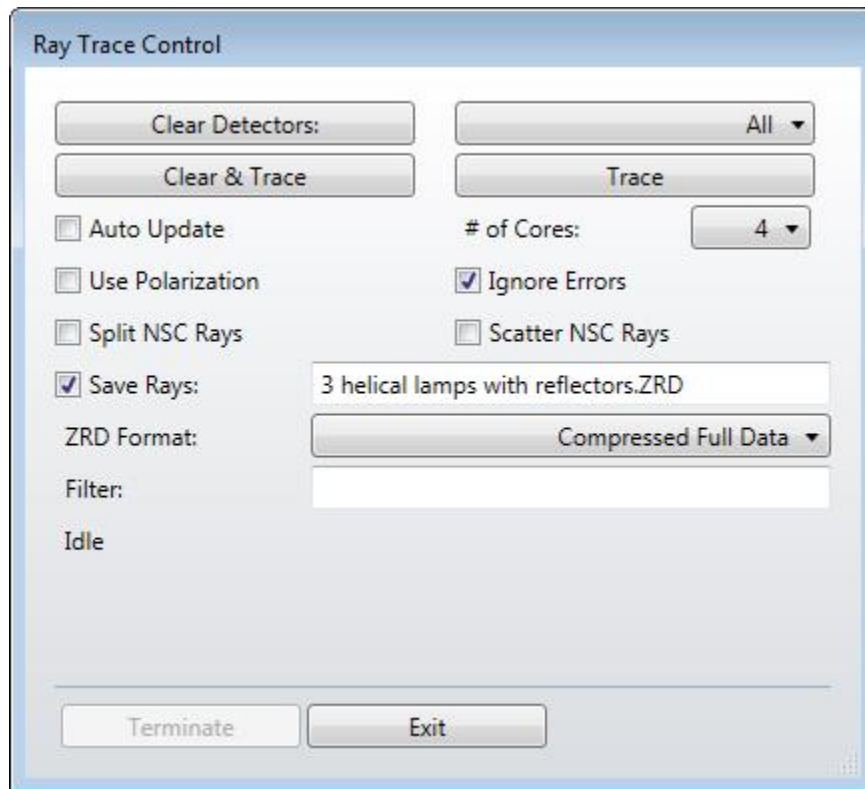
Clear Detectors Select either "All" or any one specific detector to clear. This resets all of the detector's pixels to zero energy.

Clear & Trace Clears the detectors selected and begins tracing rays. See the Clear Detector and Trace settings for details.

Trace Begins tracing rays from all defined sources. Once the trace begins, only the Terminate and Auto Update controls are available. See the discussion below on other effects of initiating a ray trace.

Auto Update If checked, all detectors will be redrawn periodically so progress may be monitored.

 # CPU's Selects the number of CPU's over which to spread the ray tracing task. More than 1 may be selected, even on a single CPU computer, in which case the single CPU will time share the multiple simultaneous tasks. The default is the number of processors detected by the operating system. Using more CPU's may speed the computation but requires significantly more memory. If Zemax reports insufficient memory errors or begins using the hard disk as a memory cache, try reducing the number of CPU's.

Use Polarization If checked, polarization will be used to determine ray energy, reflection, transmission, absorption, and thin film effects. If Split Rays is checked, Use Polarization will be checked automatically, as it is required for accurate ray splitting.

Split Rays If checked, rays will split according to the interface properties as described in "Ray splitting" .

Scatter Rays If checked, rays will scatter according to the scatter properties of the interface.

Ignore Errors If checked, errors generated by ray tracing are ignored, and any errors that occur will not terminate the ray trace. This should be turned on with caution, since errors may indicate a serious flaw in the system being traced. However, some perfectly good systems occasionally have a few rays fail because of finite computer precision or other minor problems. If the "lost energy (errors)" given at the end of the ray trace is small relative to the total power of all sources defined, then the few rays which trigger errors can be safely ignored. See "Lost energy" .

Save Rays If checked, ray data will be saved to the file name specified. The file name only should be provided with the proper extension and no folder name. See also "Filter" below.

Filter   This option is only available if "Save Rays" is checked on. If the filter is not left blank, then the defined filter will be applied to each ray tree traced before it is saved to a file. Only if the ray tree passes the filter will the ray tree be saved. This may substantially reduce the amount of data saved, and therefore may reduce the resulting ZRD or DAT file size. For a description of the filter syntax, see "The filter string" .

Discussion

There are some important considerations to understand when performing a Monte Carlo ray trace. When the detectors are cleared, all the energy counts in each pixel are reset to zero. Once the Trace button is selected, Zemax will trace exactly the number of specified analysis rays for each source listed in the NSC Editor. Each source has an associated power in watts or other units ("Analysis Units"). The initial intensity of each ray from a given source is the total power divided by the number of rays to be traced. For example, for a 1 watt source tracing 1,000,000 rays; each ray initially has 1.0E-06 watts of power associated with it.

If the ray tracing is terminated before all the rays are traced; the total power and peak power as listed on the detector objects will be inaccurate; although the ray distribution will be accurate for the number of rays traced. The analysis cannot be paused and resumed; so once a ray trace is terminated, selecting Trace again will start a new trace.

If the detectors are cleared, then Trace is pressed, the ray trace completes, and Trace is pressed again; all detectors will show the total power from both traces. Zemax has no reliable way of knowing how many times rays have been traced between detector clearings; since sources may be added, deleted, or modified at any time.

For accurate power computations, the detectors should be cleared before tracing; then a single trace should be performed without early termination.

Saving ray data to a file

If the Save Rays option is checked, some or all of the traced ray data may be saved to a file. The nature of the data saved depends upon the file name provided.

If the file extension is ZRD, the complete (optionally filtered) ray data will be saved. The folder the file will be placed in is the same as the lens file being traced. The ZRD format is used by Zemax in the ray data base viewer (see "The Ray Database Viewer"). For details on the ZRD format, see "Ray database (ZRD) files" .

If the extension is DAT or SDF, then only rays (optionally filtered) that strike a particular object are saved to a source file. This format is used by Source File objects (see "Source File"). To use the source file format, the file name must be of the form n-filename.DAT or n-filename.SDF where n is an integer corresponding to the object number desired. In this case, the filename.DAT or filename.SDF (with the n- part removed) will be created listing rays that hit object n. For example, the file name "11-ReferenceObject.DAT" will save all rays that hit object 11 into the file "ReferenceObject.DAT". The file will be placed in the folder <objects>\Sources\Source Files (see "Folders"). If the file extension DAT is used, the ray data is stored in the older "flux only" format and the wavelength data is lost. This results in smaller files and is recommended for monochromatic sources only. If the file extension SDF is used, the ray data is stored in the newer "spectral color" format, and wavelength data is stored with every ray. For more information on source file formats see "Binary Source File format" .

Saving rays slows down the calculation significantly, as writing data to a file is much slower than the ray tracing. For this reason the Save Rays feature should only be used when the data is needed for subsequent analysis.

Lost energy

When tracing some NSC systems, not every ray can be traced successfully. When a ray trace is terminated, the energy associated with the ray is called "lost energy". The lost energy is measured in source units (see "Source Units").There are different causes for terminating the trace of a particular ray:

- If a ray falls below the minimum energy threshold for tracing, the ray will be terminated. The minimum relative and absolute energy thresholds are described in "Non-Sequential" . The ray trace control reports this type of lost energy as the energy lost due to threshold settings.

- Sometimes geometry errors or round-off errors lead to conditions where the ray can no longer be traced. Geometry errors are usually caused by poorly formed solids or improper nesting of objects. For some rays, correct intercept or refraction data cannot be computed. This may happen when iteration to a surface fails, or the coating data cannot be computed. Rays that hit the maximum number of object intercepts, nesting levels, or segment levels, are also terminated with an error condition. The ray trace control reports these types of lost energy as the energy lost due to errors.

- Rays that strike the edge of an object may be terminated if the local refraction and/or reflection cannot be computed accurately. A ray is considered to strike the edge of an object if the intercept point lies within a threshold distance to the edge, or if two or more objects are struck at the same point and the objects have non-parallel normals at the intercept point. For most objects, the threshold distance is approximately the glue distance in lens units. Along the edge of a faceted object, the threshold distance is approximately the glue distance (interpreted here as a dimensionless number) multiplied by the dimensions of the facet. Note that rays that strike very close to the edge of a part have no useful physical interpretation anyway; an actual object would diffract (or scatter) all energy incident within a few wavelengths of the edge. Rays that miss all objects, or are absorbed by either coatings, bulk transmission, or detectors, are also terminated. Zemax absorbs all these rays without error or report, as the energy is not considered lost because the energy is known to be absorbed or propagated out of the system and is thus accounted for.

If the fraction of lost energy is relatively small, lost energy can be ignored. Even well designed and accurate optical models will occasionally have spurious ray errors. Rays that strike within edges or seams between parts and are thus absorbed will usually represent an insignificant fraction of the total energy in the system. If the lost energy is unacceptably high, then the settings (see for example "Minimum Relative Ray Intensity") or the object geometry needs to be checked. For assistance in studying geometry errors, see "Create Source Ray From Last Geometry Error Tool" .

Lightning Trace

The LightningTrace Control is used to launch and trace a specific set of rays from a discrete mesh for analysis using defined sources, objects, and detectors. The controls on the dialog box are described below.
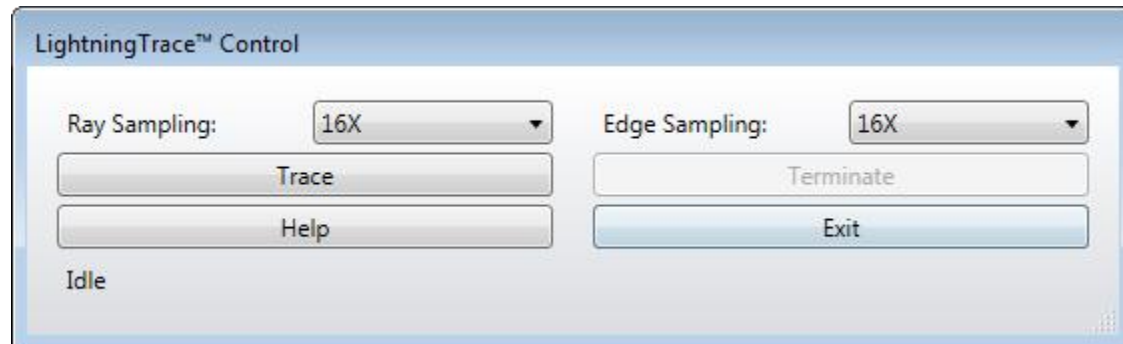
Ray Sampling   Sampling used to define the initial mesh for ray launch.

Edge Sampling   Sampling used to define the accuracy with which the mesh resolves the edges of objects.

Trace   Begins tracing mesh rays from all defined sources.All detectors are automatically cleared (i.e. all pixels on all detectors are reset to have zero energy) prior to raytracing.

Discussion

This feature uses an advanced ray interpolation technology (patent pending) to quickly estimate illumination patterns without the need for tracing millions of rays. The intended application is very fast determination of approximate lighting patterns, allowing initial setup and design of illumination systems to proceed much faster than by using conventional ray tracing (see "The Ray Trace Control").

When using LightningTrace, a specific set of rays are launched and traced from each defined source. The launch directions are chosen along a mesh which encapsulates the source. The resolution of this mesh is defined by the Ray Sampling input, and varies from "Low (1X)" to "1024X" (note that as the mesh size doubles, the number of mesh rays

quadruples). All rays launched from each mesh start at the same position, corresponding to the center position of the corresponding source. Thus, LightningTrace neglects the size and spatial distribution of each source, and represents each source by its far-field description.

As rays are traced through the system, Zemax automatically refines the mesh as needed to ensure that a sufficient number of rays pass through each object on the way from the source to any detectors. The user may control the level of refinement made at the edges of objects via the Edge Sampling input. This input also varies from "Low (1X)" to "1024X".

Rays which leave the mesh are traced in the usual fashion. Mesh rays may only follow the specular ray path, however; splitting and scattering of these rays is not supported. Polarization effects are also not accounted for in tracing the mesh rays. Finally, while the overall color (i.e. Tristimulus values) of each source will be retained during ray tracing, including the variation of color with far-field angle, the spectral distribution of each source is neglected, and an average wavelength for the source is used to trace the mesh rays.

The initial mesh of rays used by the LightningTrace analysis (i.e. the mesh that is generated prior to any automatic refinements made by Zemax) may be visualized in the NSC 3D Layout and the NSC Shaded Model (see "NSC 3D Layout" and "NSC Shaded Model"). Note that the intensity of each ray leaving the mesh is not constant; ray intensities in any angular region are given by the far-field intensity of the source in that region.

Results of the LightningTrace analysis may be observed on Detector Rectangle, Detector Color, and Detector Polar objects (see the "NSC Detectors" chapter). To observe the results on any of these detectors, the detector must either be a terminal detector in the system (i.e. rays do not interact with any other objects after hitting the detector) or the material for the detector must be set to "ABSORB". In addition, only the spatial data will be available when viewing the results on a Detector Rectangle or Detector Color object, while angular data can be obtained by using a Detector Polar object. The results can be viewed on the Detector Viewer (see "The Detector Viewer") in the usual fashion.

Nearly all sources are supported by LightningTrace, including Source Files generated from RSMX measure- ments (see "Generate Radiant Source Model Rays"). However, sources defined via the Source DLL, Source Imported, and Source Object objects are also not supported by LightningTrace. For such sources, it is very likely that the size and spatial distribution of the source will be important to the system design. Thus

LightningTrace will be inappropriate for modeling a system containing such sources. If a system contains any of the sources not supported by LightningTrace, that system cannot be analyzed with LightningTrace.

Suggestions for use

The main benefit of LightningTrace is speed. Since rays are only traced from a mesh, the number of rays traced through the system will generally be much smaller than used during conventional ray tracing. The response of the system for regions between the mesh are computed using an advanced ray interpolation technology to estimate the behavior of the mesh rays. As such, LightningTrace represents an approximation of the system performance, and in some cases could lead to artifacts (such as hot spots) in the results. A common example of hot spot generation occurs when LightningTrace is used in symmetric (e.g. on-axis) optical systems with results being viewed on detectors with odd numbers of pixels. However, for many lighting and illumination systems the approximations used by LightningTrace are valid. In such systems, ray tracing and optimization (which is available via the NSLT operand; see "Optimization operands") can be orders of magnitude faster when using LightningTrace as compared to conventional ray tracing.

Systems for which LightningTrace will work well are those systems in which the size and spatial distribution of all sources is not important to the system design, either because the sources themselves are sufficiently small and/or are sufficiently far enough away from all secondary optics. For example, LightningTrace would be well- suited for modeling systems containing very small LED sources with complex far-field distribution patterns. Thus, LightningTrace can be an ideal tool for designing LED lighting systems, examples of which include architectural lighting or street/roadway lighting (for a description of tools to aid in designing roadway lighting, see "Roadway Lighting").

LightningTrace is not the appropriate tool to use for critical illumination systems (in which sources are to be imaged onto detectors) or systems in which the size and spatial distribution of any source is important (such as an LED collimator or beam expander). In addition, LightningTrace will not be accurate for systems containing diffusers (or other scattering components) or for systems with significant dispersion (e.g. due to the presence of diffraction gratings). Finally, LightningTrace cannot be used to model perfectly collimated beams in an optical system. All of the above systems should be designed using conventional ray tracing.

For the class of systems for which LightningTrace is appropriate (e.g. LED lighting), LightningTrace provides users with a fast, virtual prototyping tool for their designs. For these systems, LightningTrace may be used very effectively with the Visual Optimizer (see "Visual Optimization") to observe the variation of system performance with design inputs in real time. A conventional ray-trace analysis, preferably using Radiant Source Models, should always be performed as the design matures to validate the LightningTrace analysis.

Detectors Group
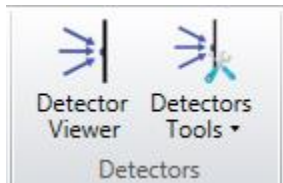
Detector Viewer

The Detector Viewer displays data from detectors in either a graphic or text format. For information on tracing rays to record data on detectors, see "The Ray Trace Control".
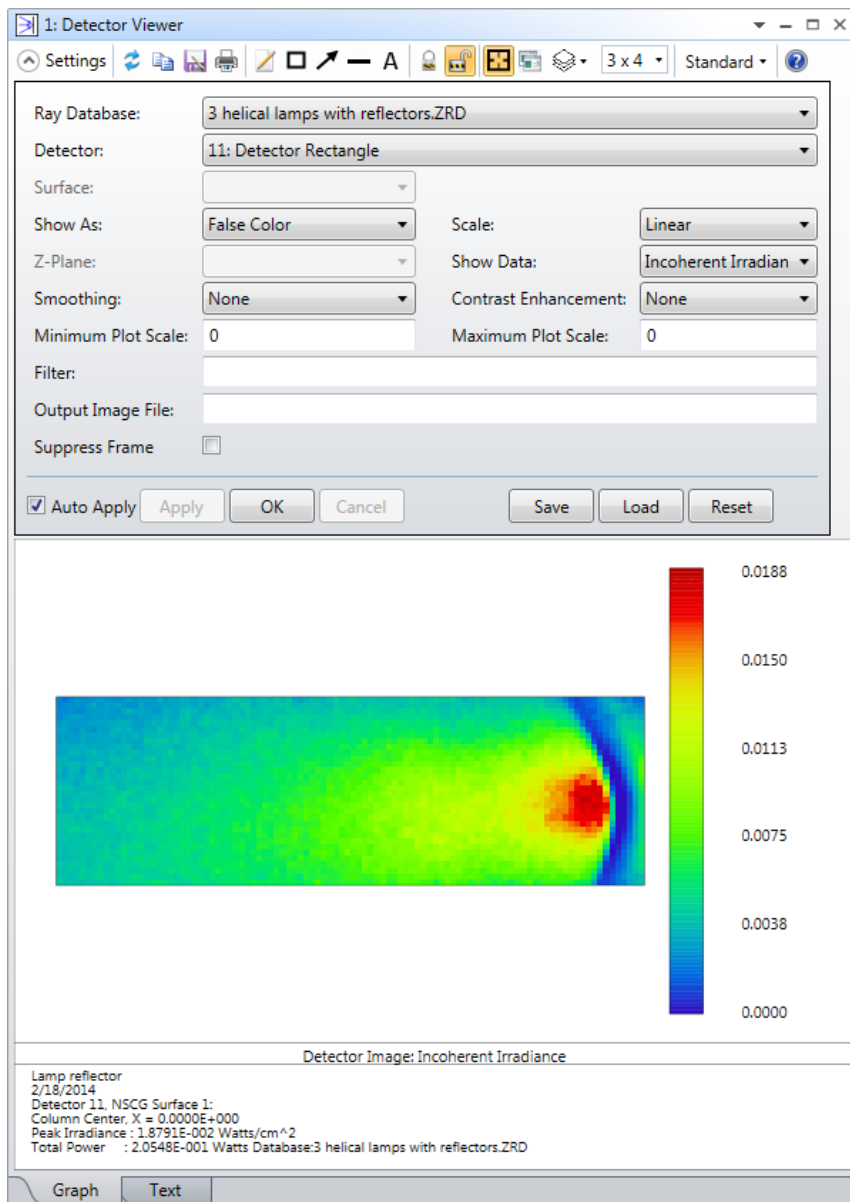
Surface Selects the NSC surface number. This is always surface 1 when in NSC mode.

Detector Selects the detector object. If the window is a text window, there is also an option to summarize all detectors.

Show As Choose grey scale or false color display, or cross section plots of rows or columns. Detector color objects support a flux vs. wavelength plot, which is only available if the option to "Record spectral data" has been selected for the detector (see the description given under the "Type tab"). Detector polar objects support either a full or half polar directivity plot, which displays normalized detected energy as a function of angle. Cross sections are not available in all text displays.

Row/Column Selects the row or column to display for cross section plots. For Detector Surface objects, see "Comments regarding Detector Surface objects" .

Angle Detector polar objects support cross sections displayed as linear in angle, or in full or half polar directivity plots. This angle control defines the angle of the plane defining the cross section through the data, relative to the positive local x axis of the detector.

When displaying directivity plots only, the Angle control name becomes Angle(s). The control then supports multiple angles separated by commas. For example, the Angle(s) list may be 0, 45, 90, 135. All the angles are then shown on the same plot. A maximum of 10 angles may be listed.

Z-Plane This control is only used by detector volume objects. The Z-Plane selects which cross section of a volume detector to display. The number of Z-Planes in a detector volume is set by the number of voxels in the Z direction. The text version of the Detector Viewer also supports an "All' option to display all cross sections as well as a table of all voxel center coordinates and data. Pressing the left and right arrow keys while viewing detector volume data will cycle through all the Z-Planes.

Smoothing Smoothing reduces pixel noise due to low sampling by averaging the data in adjacent pixels. The data in every pixel is replaced by the average of the data in the pixel and the neighboring pixels. This averaging algorithm is applied the number of times specified by the smoothing parameter.

Show Data See discussion below for important information.

Ray Database If none is selected, the ray data currently stored on the detectors is used. Otherwise, the previously saved ray database data is used.

Filter A filter string may be applied to select only rays with particular properties. This option is only available when using a ray database. For information on the filter string syntax, see "The filter string" .

Scale Choose linear or logarithmic scales. See "Max/Min Plot Scale" below.

Max/Min Plot Scale If non-zero values are provided, the plot will be scaled to the specified min and max data range. If these values are both zero, the default scaling will be applied.

If the Scale (see above) is Logarithmic, the Min Plot Scale setting is ignored. If the Max Plot Scale setting is greater than zero, then the next greater power of 10 is used as the maximum log scale.

Contrast Enhancement By default, no scaling of the data is done. If Contrast Enhancement, a non-linear weighting function is applied to the pixel data to exaggerate the less intense pixels. This can be useful to bring out fine detail in the image, however, the visual appearance may be misleading since dim regions will be brighter in the image than they would really be. This option is only supported for detector color and rectangle objects, and if log scaling is not used.

Output Image File If a file name is provided, the detected image will be saved to the file. The file name should include the full path, or the default path will be the same as for IMA files. The file extension should be BMP, JPG or PNG. If the file extension is not recognized JPG will be used. This option is only supported for detector color and rectangle objects. The exported image is at the full resolution of the detector, not the screen, and will use the same number of pixels in the vertical and horizontal directions as the detector does. However, the pixels in the exported image are assumed to be square, so if the detector pixels are not square some distortion of the aspect ratio of the image will be introduced.

Suppress Frame If checked, the frame is not drawn and the entire window is used to display the image. This feature is not supported for polar detectors.

Discussion

Once ray data has been recorded on the detectors, or from rays stored in a ray database, the Show Data option determines what aspect of the data is displayed as follows:

Incoherent Irradiance: This is incoherent power per area as a function of spatial position on the detector. The power of each ray that strikes the same pixel is summed without regard to the phase of the ray. This option becomes Incoherent Illuminance when using photometric rather than radiometric units.

Coherent Irradiance: This is coherent power per area as a function of spatial position on the detector. The complex amplitude of each ray that strikes the same pixel is summed, keeping track of the real and imaginary parts separately considering the phase of each ray. The final amplitude is then squared to yield the coherent power. This option becomes Coherent Illuminance when using photometric rather than radiometric units.

Coherent Phase: The phase angle of the complex amplitude sum used in Coherent Irradiance.

Radiant Intensity: The power per solid angle in steradians as a function of incident angle upon the detector. This option becomes Luminous Intensity when using photometric rather than radiometric units.

Radiance (Position Space): The power per area per solid angle in steradians as a function of spatial position on the detector. Since it is not possible to display the variation in angle space and position space at the same time, this plot is effectively identical to the Incoherent Irradiance display, with the data values divided by two pi. Note this scaling assumes the detector is collecting data over a hemisphere, no matter what angular range the detector actually collects rays over. This option becomes Luminance when using photometric rather than radiometric units.

Radiance (Angle Space): The power per area per solid angle in steradians as a function of incident angle upon the detector. Since it is not possible to display the variation in angle space and position space at the same time, this plot is similar to the Radiant Intensity display, with the data values divided by the product of the area of the detector and the cosine of the angle at the center of the pixel. This option becomes Luminance when using photometric rather than radiometric units.

Incident Flux: For detector volumes, the incident flux is the flux entering each voxel, measured in source units

(watts, lumens, or joules).

Absorbed Flux: For detector volumes, the absorbed flux is the flux absorbed within each voxel, measured in source units (watts, lumens, or joules).

Absorbed Flux/Volume: For detector volumes, the absorbed flux/volume is the flux absorbed within each voxel, measured in source units (watts, lumens, or joules) per unit volume measured in cubic lens units.

Flux vs. Wavelength: For detector color objects, the spectral distribution of the flux incident on the detector. The plot will be shown as a bar chart. To obtain the spectral distribution on any object (other than a detector color), use the Flux vs. Wavelength analysis (see "Flux vs. Wavelength").

Detector Tools
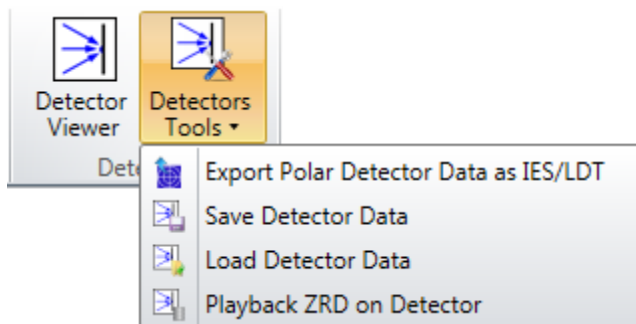
Perform various functions on detector data.

Export Polar Detector Data as IES/LDT

Saves intensity data recorded on a polar detector into an IESNA or EULUMDAT file.
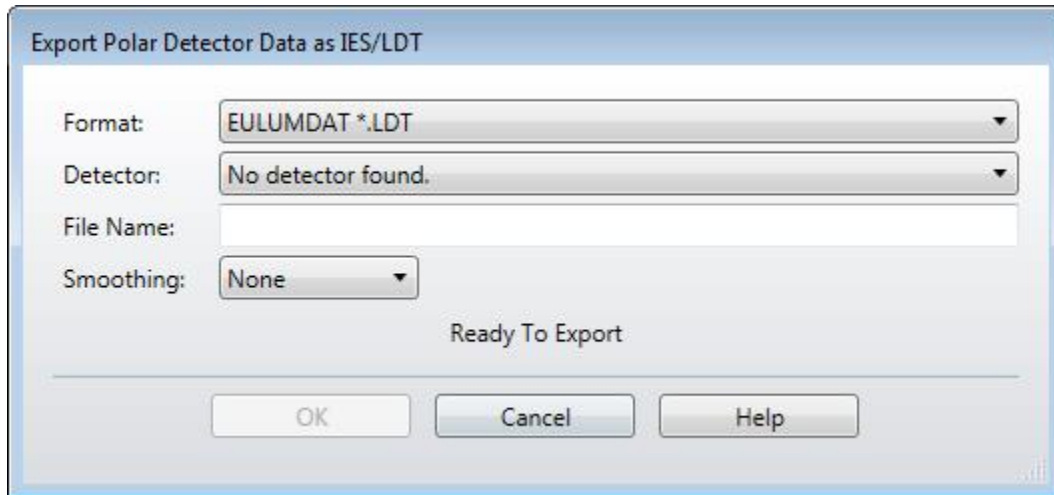
Format Data may be saved in either the IESNA or EULUMDAT format. For more information on the IESNA format supported by Zemax, see "Source IESNA File" . For more information on the EULUMDAT format supported by Zemax, see "Source EULUMDAT File" .

Detector The object number of the Detector Polar from which the intensity data will be saved.

File Name The name of the file to be saved. IESNA files will be saved in the

<objects>\Sources\IESNA folder, while EULUMDAT files will be saved in the

<objects>\Sources\EULUMDAT folder. For more information on the <objects> folder, see "Folders" .

Smoothing The level of smoothing to use in processing the data. The smoothing algorithm is described in "The Detector Viewer" .

Description

Intensity values recorded on a Detector Polar object may be saved into a file format commonly used for describing photometric data from a real lamp. These data represent the angular distribution of the emission only; for more information, see "Detector Polar object" .

Note that the IESNA specification requires polar angle data be defined from 0 to 90 or 0 to 180 degrees. As such, it is recommended that this tool only be used with polar detectors whose maximum angle is defined to be 90 or 180 degrees. If this is not the case, Zemax will issue a warning and pad the exported data with zeros out to 90 or 180 degrees.

Save Detector Data

Saves the data for the detector currently selected in the editor. The only supported detectors are the Detector Rectangle, Detector Color, Detector Polar, and Detector Volume objects. The file format and data stored is specific to each detector type, and the distinct extensions DDR, DDC, DDP, and DDV are used to denote the different file formats. For information on the file formats, see "Saving and Loading Detector Data".

Load Detector Data

Loads data previously saved in a DDR, DDC, DDP, or DDV file onto the detector currently selected in the editor. If the detector size, angular range, pixel count, or other property does not match the target detector, then the detector properties are automatically modified to match that of the incoming data. If the units of the stored data do not match the current units, an error is issued and the detector is not modified. For information on the file formats, see "Saving and Loading Detector Data" .

Playback ZRD on Detector

Reload saved ray database data onto detectors.
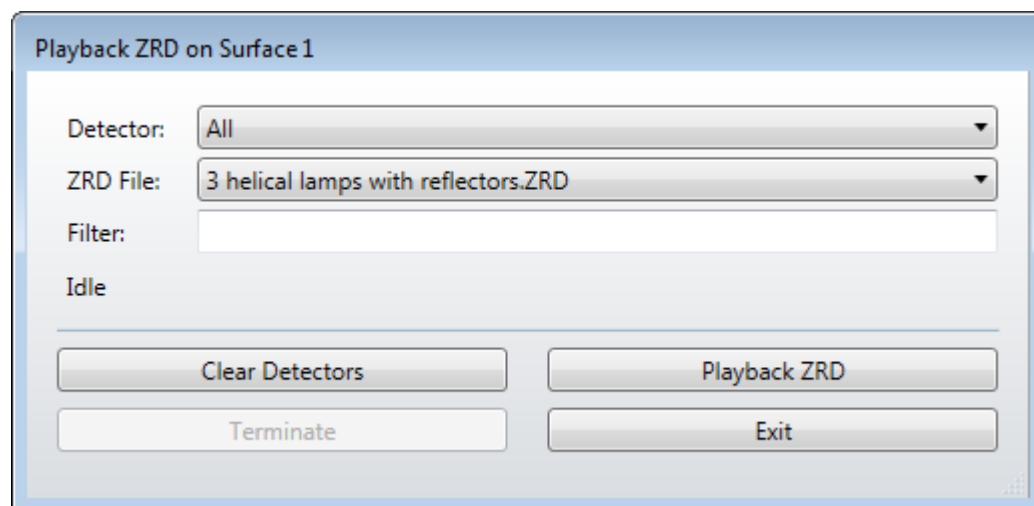
Detector   Data may be loaded onto an individual detector or all detectors simultaneously.

ZRD File   The name of the file to be saved. IESNA files will be saved in the

<objects>\Sources\IESNA folder, while EULUMDAT files will be saved in the

<objects>\Sources\EULUMDAT folder. For more information on the <objects> folder, see "Folders" .

Filter   An (optional) filter string used to isolate specific rays in the file. For more information, see "The filter string" .

Discussion

Ray trace results are often saved to a ray database (ZRD) file for further analysis. Using this tool, the results from a saved ZRD file may be reloaded onto a particular detector or onto all detectors in the system. A corresponding ZPL keyword (ZRDPLAYBACK) is available to reload the ZRD file within a macro. Using this keyword, data saved to a ZRD file may then be filtered (using filter strings) as a part of post-processing. For more information see "ZRDPLAYBACK" .

Raytrace Analysis Group
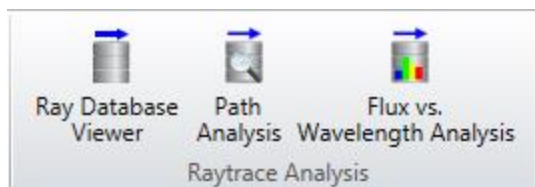
Ray Database Viewer

The Ray Database Viewer reads in a previously saved ZRD file, and displays the data in a text format. For information on creating a ZRD file, see the previous section.
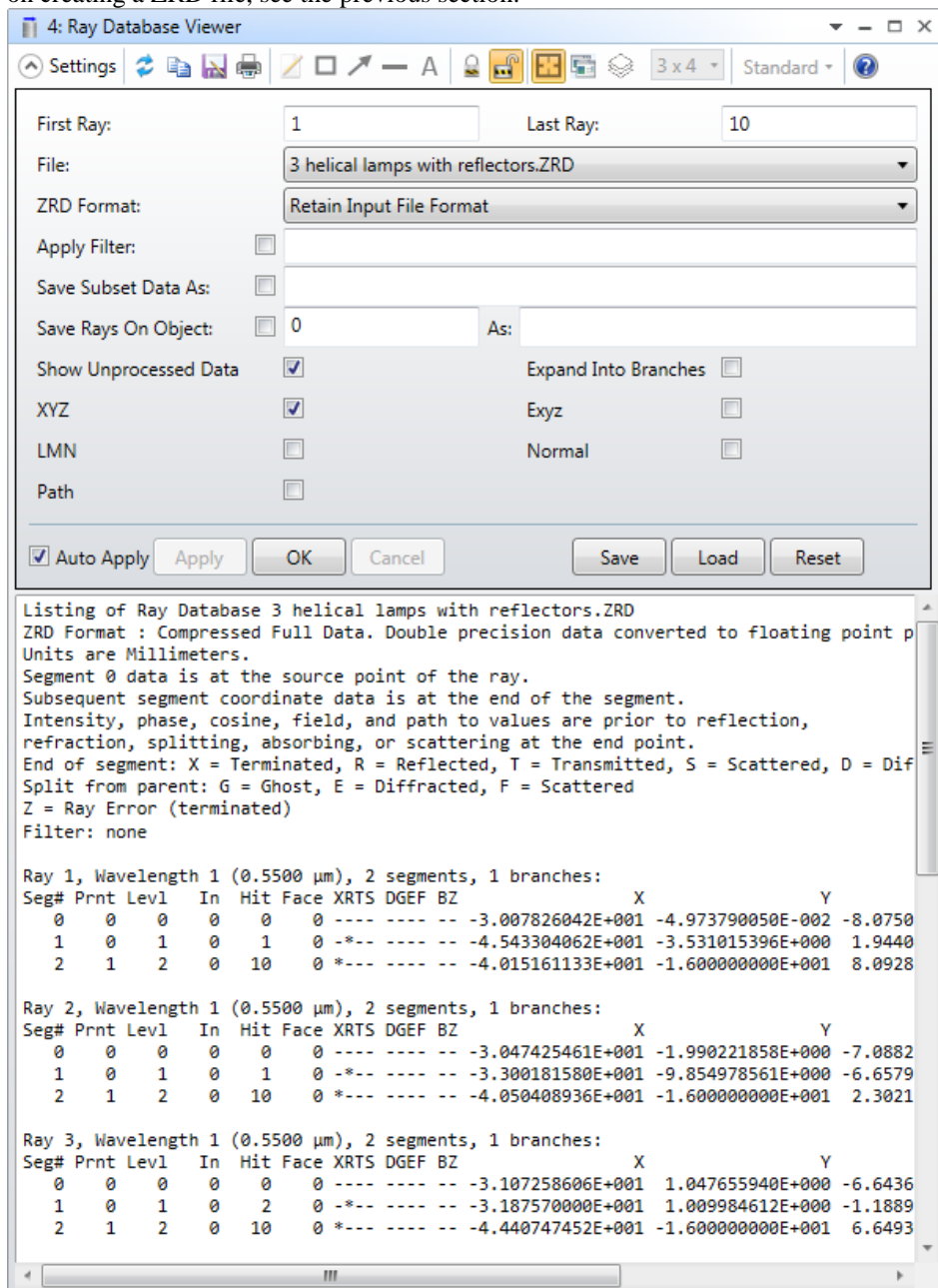
File   The name of the ZRD file to use. Zemax will look in the folder where the current lens is saved, and list all files with the ZRD extension found there.

Show Unprocessed Data If checked, the complete data tree for each ray traced will be displayed.

Expand into Branches If checked, each branch of the tree will be expanded to show a single complete path from source to ray termination.

XYZ If checked, the global X, Y, and Z coordinate data of the ray segment-object intercept point will be displayed.

LMN If checked, the global L, M, and N direction cosine data of the segment will be displayed.

Normal If checked, the global Nx, Ny, and Nz object normal vector components at the ray segment-object intercept point will be displayed.

Path If checked, the path length in lens units from the parent segment is displayed. This is the physical, not the optical path length. Zemax will also show the phase in radians, but only for segments striking a detector surface.

Exyz If checked, the global electric field in the X, Y, and Z directions will be displayed.

First/Last Ray The first and last number of the rays to display. These controls are very useful for limiting the length of the displayed data. If the subset data is to be saved in a new ZRD file, only rays within this range will be included. The value -1 may be used for the Last Ray, which indicates the last ray defined in the file. If -1 is used for the Last Ray, text output will be suppressed.

Apply Filter If checked, the filter string entered will be used to reduce the amount of data displayed. The filter string is also used to determine the subset of the segments which are saved to a new ZRD file. For information on the filter string syntax, see "The filter string" .

Save Subset Data As If checked, the data displayed which passes the filter string will be saved in ZRD format in the file name specified. The file name should end in the extension ZRD, with no path provided. The path used will be the same as the current lens file. To prevent accidental overwrite of the subset data file, this control is automatically "unchecked" after every save.

ZRD Format This control selects the format to be used for the ZRD file created if "Save Subset Data As" is checked. For information on ZRD formats, see "Ray database (ZRD) files" .

Save Rays on Object "n" As If checked, ray segments which hit the specified object number and which pass the defined filter (if any) will be saved as source rays in a binary source file format of the specified name. Note the ray position data is saved in global coordinates. This file may subsequently be used as a source file. For information on the ray source file format, see "Source File" . The file name should end in the extension DAT (for flux only monochromatic data) or SDF (for polychromatic data), with no path provided. The folder used will be <objects>\sources\Source Files.

Only rays within the range of the first/last ray setting are saved. To save all rays that hit the specified object from the ZRD file, choose 1 for the first ray and any integer greater than the number of rays in the ZRD file for the last ray. If the total number of rays exceeds 1000, then printing of the individual ray data in the ray database report will be suppressed to speed up the process. To see the text listing, choose "Update" again from the text window menu bar. The second update will show the text data because the "Save Rays On Object" box is automatically unchecked after every save.

Discussion

The text display for the ray database viewer can be quite long, so it is best to limit the amount of data displayed by using the first/last controls and the filter string.

A typical unprocessed ray listing looks like this, with some columns removed for brevity:

Ray 1, Wavelength 1:

Seg# Prnt Levl In Hit XRTS DGEF BZ   X   Y   Z

0   0   0   0   0   ----   ----   --   0.0000E+000   5.0000E+000
         0.0000E+000

Seg#	Prnt	Levl	In	Hit	XRTS	DGEF	BZ	X	Y	Z
1	0	1	0	2	-**-	----	--	9.0000E+000	0.0000E+000	3.0870E+000
2	1	2	0	0	*---	-*--	--	7.2002E+000	0.0000E+000	2.7044E+000
3	1	2	2	2	-**-	----	--	1.2838E+001	0.0000E+000	2.5468E+000
4	3	3	2	2	-**-	-*--	--	9.0000E+000	0.0000E+000	9.5506E-001
5	4	4	2	2	-**-	-*--	--	1.2988E+001	0.0000E+000	-6.9886E-001
6	5	5	2	2	-**-	-*--	--	9.0000E+000	0.0000E+000	-2.0350E+000
7	6	6	2	2	--*-	-*--	--	1.2730E+001	0.0000E+000	-3.2848E+000
8	7	7	0	3	*---	----	--	1.8400E+001	0.0000E+000	-5.7054E+000
9	6	6	0	0	*---	----	--	7.3797E+000	0.0000E+000	-2.9069E+000
10	5	5	0	3	*---	----	--	1.8400E+001	0.0000E+000	-4.2858E+000
11	4	4	0	0	*---	----	--	7.4900E+000	0.0000E+000	-9.6429E-002
12	3	3	0	3	*---	----	--	1.8400E+001	0.0000E+000	9.6510E-001

"Seg#" is the segment #, and is sequential from 0 to the total number of segments the ray generated.

"Prnt" is the parent number, which indicates which prior segment generated that particular segment. For example, the parent of segment 11 is segment 4, whose parent is 3, whose parent is 1, whose parent is 0. Segment 0 is the source segment, and the starting coordinates indicate the location of the source point. Any segment can be traced back to the source in this manner.

"Levl" is how many segments away from the source ray the segment is. For segment 11, there are 4 segments back to the source (4, 3, 1, and 0).

"In" indicates what object number the segment is traveling through.

"Hit" indicates the object number struck at the end of the segment. The source segment has a hit number of

0, since the XYZ coordinates listed are the starting and not the terminating coordinates of the segment. For all other segments, the XYZ coordinates and other data are at the segment end on the "hit" object. If the hit object is 0, the ray missed all objects and is terminated.

"XRTS DGEF BZ" is a quick indicator of what happened to the segment. X means terminated, R reflected, T transmitted, S scattered, D diffracted, and B bulk scattered. All these values indicate what happened at the end of the segment, when the segment struck the object "hit". The values G, E, and F indicate the segment was a ghost, diffracted, or scattered, respectively, from the parent segment's "hit" object number, so these events occurred atthestart of the segment. The symbol Z indicates a ray error occurred and the ray was terminated. Note more than one event may occur for any ray segment.

The other data displayed includes the global XYZ coordinates of the end of the segment, the global normal vector coordinates of the object struck at the end of the segment, the ray intensity, the object comments, and possible other data as this feature is expanded. Object comments include which object the ray is currently inside of, whether the ray

underwent bulk scattering inside of an object, and whether the wavelength for the ray was modified during a bulk scattering event (wavelength modification during bulk scattering is described in detail in "Bulk Scatter tab").

Path Analysis

This feature is only available in the Premium edition of Zemax.

This feature groups rays from a ZRD file according to the path the rays took through the system. Rays that took the same path, in terms of the order different objects were interacted with, are summed so the total flux in each path can be determined. This feature also may automatically generate path filters so the groups of rays may be studied separately.
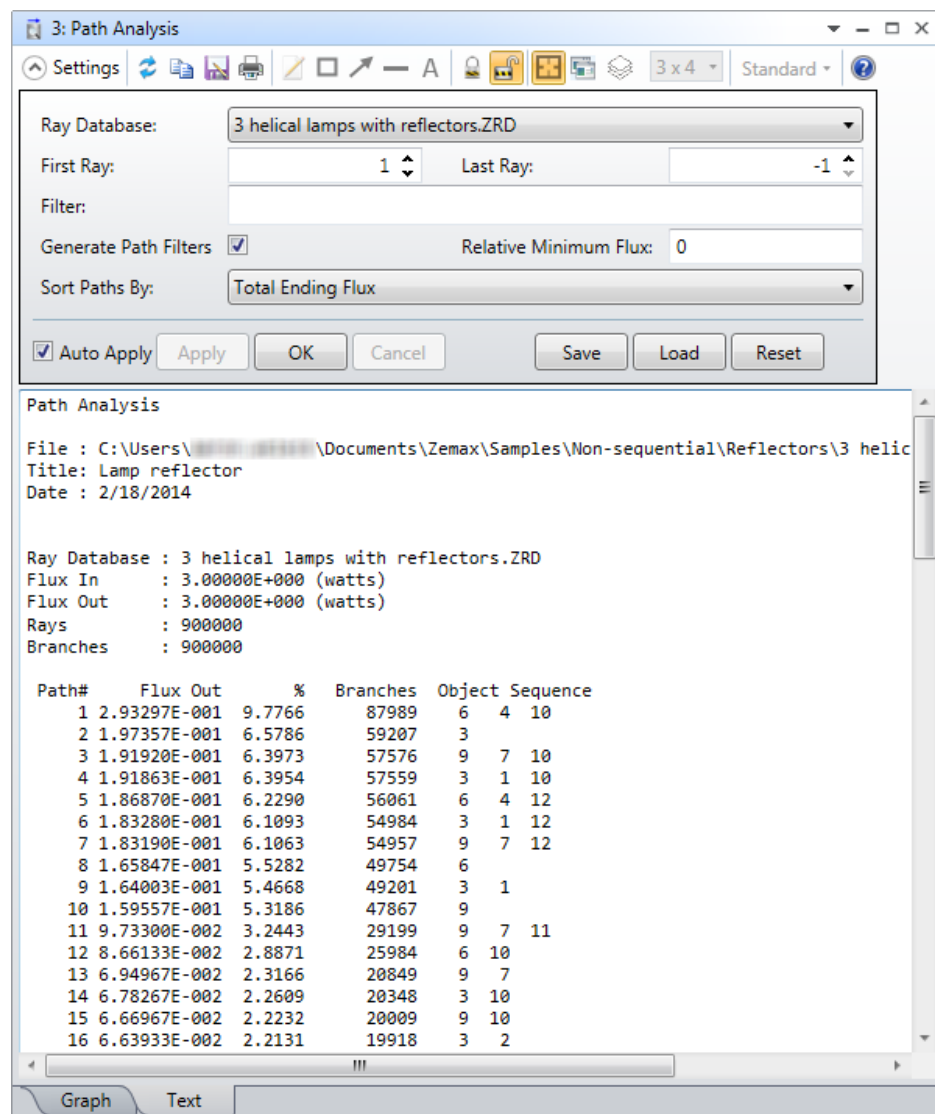
Ray Database   Selects the existing ZRD file to analyze.

First/Last Ray   Choose the ray range. To use all rays, set the First ray to 1 and the last ray to -1.

Filter   The filter string. See "The filter string" .

Generate Path Filters   If selected, path filters for all possible paths are automatically generated.

Relative Minimum Flux The relative flux is the amount of energy at the end of a branch relative to the starting energy at launch. If this value is less than the specified minimum, then the branch is added to the "uncategorized" sum.

Discussion

This feature reads all or some subset of the rays in the selected ZRD file, and filters the rays according to the specified filter string, if any. Then, each branch of each ray is analyzed, and the order of object interaction is determined. This order is called a "path". If a ray starts at source object 3, then hits the front of object 7 and then the back of object 7, then reflects off object 9, the path is 3, 7, 9. Note the interaction with object 7 twice is considered to be a single interaction, so the path is not 3, 7, 7, 9. The same object number will never appear twice consecutively in a path. There is a maximum number of objects allowed in any one path, and this limit is currently 100. All paths with more than this number of object interactions are considered to have the same path.

All branches that have the same path are grouped together. The flux (always in watts, regardless of the source units) at the end of the path is summed for all branches having the identical path. The list of all possible paths is generated and sorted in the order of total ending flux. This data is summarized in a report listing the path #, flux out, relative flux out as a percentage of the total flux in for all rays, the number of branches with that same path, and then the object sequence in the path. Branches that do not fall into any path because of a Relative Minimum Flux requirement, or because the maximum number of unique paths (currently 10,000) is exceeded, are added to the catch-all "uncategorized" path.

Only one ZRD file can be active with the Path Analysis feature at one time, however many path analysis windows are open.

If "Generate Path Filters" is checked, this feature will create a path filter for each path defined, and this filter will be named _1 for path 1, _2 for path 2, etc. These path filters may subsequently be used in any analysis feature that supports filter strings to isolate the selected path. The generated path filters are automatically saved with the lens file.

There is a handy shortcut for using path filters when looking at the NSC 3D Layout, NSC Shaded Model, or Detector Viewer plots: press the spacebar to increment the path filter number (from 1 to 2 or 5 to 6, etc). Press Ctrl-Space to decrement the path filter number. Also, holding down the Shift key while pressing the space bar will increment or decrement the path filter by 10.

Path filters may also be used in combination with other filters, whether explicit or named. For example, the following is a valid filter that selects only ghost rays off object 11 with the specified path: "_7 & G11". For more information see "The filter string" and "Named Filters" .

Flux vs. Wavelength Analysis

This feature selects all rays from a ZRD or SDF ray database file, and sums the intensity of rays striking a specific object as a function of wavelength.
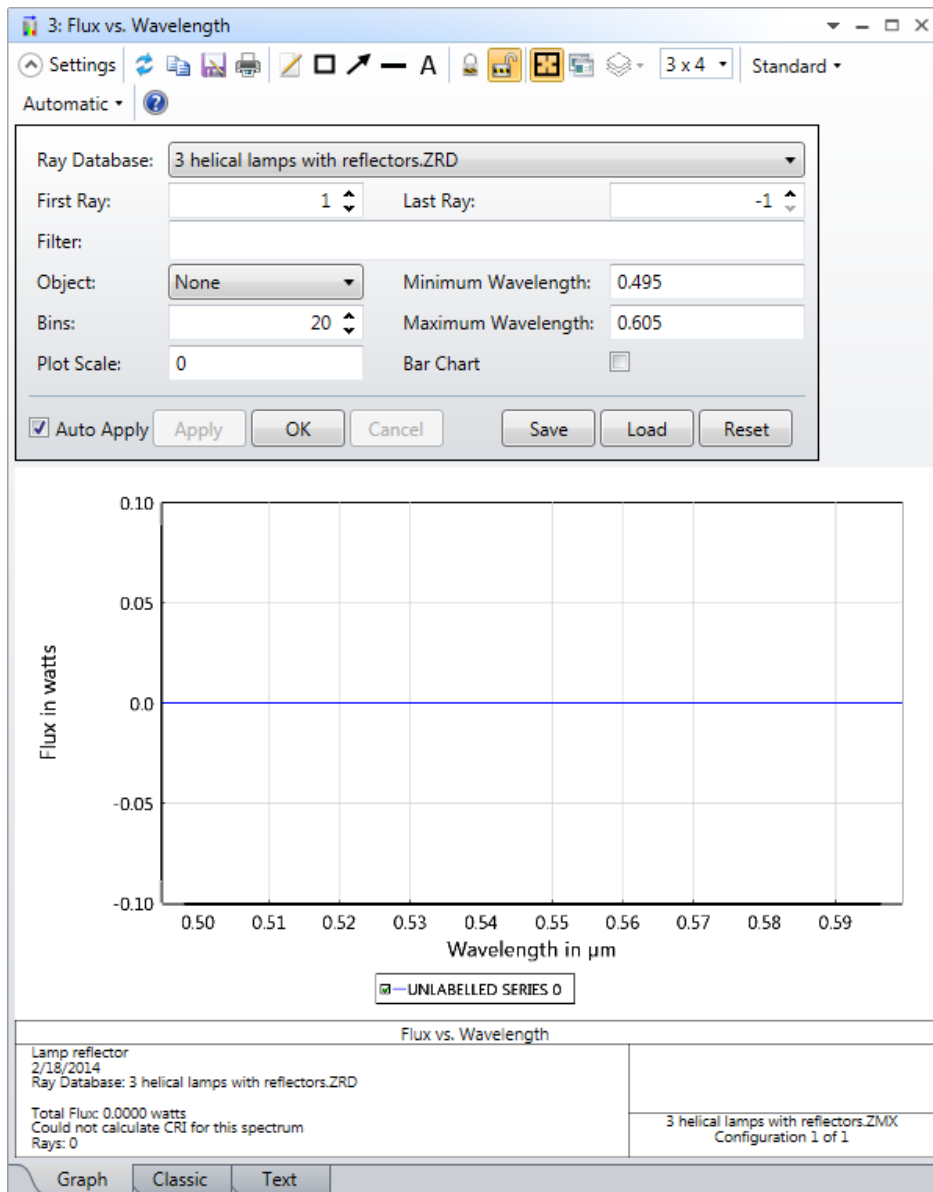
Ray Database   Selects the existing ZRD or SDF file to analyze.

First/Last Ray   Choose the ray range. To use all rays, set the First ray to 1 and the last ray to -1.

Filter   The filter string. See "The filter string" .

Object   The object the rays strike. A specific object must be selected, and the intensity for any segment that strikes the object will be added to the sum for the associated wavelength.

Bins   Because the ray wavelengths are discrete, the sum for each wavelength must be over a finite bandwidth. This setting determines how many wavelength bins are used to span the range from minimum to maximum wavelength.

Minimum/Maximum Wavelength The lowest and highest wavelength. Rays outside this wavelength range are ignored.

Plot Scale   The maximum scale for the plot. Use zero to choose a default scale.

Bar Chart   If selected, the data will be displayed as a bar chart.

Discussion

Because the wavelengths for each segment are discrete and are only determined by the rays in the ray database file, this plot can take on a noisy look with lots of data "spikes". To minimize the noisy nature of this computation, the number of rays traced should be large, and the number of bins kept reasonably small.

Polarization Group

Coatings

For more information on coating calculations, see "Polarization Analysis" in the Technical Reference section.

Reflection vs. Angle

Computes the S, P, and average polarization intensity coefficients for reflection for the specified surface as a function of incident angle.

Min Angle The minimum angle of incidence to plot. This defines the left edge of the plot.

Max Angle The maximum angle of incidence to plot. This defines the right edge of the plot.

Min Y The minimum y value to plot. This defines the bottom edge of the plot.

Max Y The maximum y value to plot. This defines the top edge of the plot.

Surface The surface number which defines the interface to use.

Object The object number if the surface is a Non-Sequential Components surface.

Face The face number which defines the interface to use, if the surface is a Non-Sequential Components surface.

Wavelength The wavelength number to use.

Reverse Direction If the selected surface is a Non-Sequential Components surface, choosing this option will reverse the incident media and substrate.

Discussion The incident angle is measured in the medium prior to the specified surface.

Transmission vs. Angle

Computes the S, P and average polarization intensity coefficients for transmission for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Absorption vs. Angle

Computes the S, P, and average polarization intensity coefficients for absorption for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Diattenuation vs. Angle

Computes the R (reflected) and T (transmitted) diattenuation for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Phase vs. Angle

For sequential surfaces, this feature computes the S and P polarization phase for reflection (if the glass is MIRROR) or for transmission (if the glass is not MIRROR) as a function of incident angle. For non-sequential objects, this feature computes the S and P polarization phase for reflection (if the material is MIRROR) or for transmission (if the material is not MIRROR) as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Retardance vs. Angle

Computes the retardance for the specified surface as a function of incident angle.

See "Reflectivity vs. Angle" for settings

Discussion The incident angle is measured in the medium prior to the specified surface.

Reflectivity vs. Wavelength

Computes the S, P, and average polarization intensity coefficients for reflection for the specified surface as a function of incident wavelength.

Min Wave The minimum wavelength to plot. This defines the left edge of the plot.

Max Wave The maximum wavelength to plot. This defines the right edge of the plot.

Min Y The minimum y value to plot. This defines the bottom edge of the plot.

Max Y The maximum y value to plot. This defines the top edge of the plot.

Surface The surface number which defines the interface to use.

Object The object number if the surface is a Non-Sequential Components surface.

Face The face number which defines the interface to use, if the surface is a Non-Sequential Components surface.

Angle The angle of incidence, in degrees, to use.

Reverse Direction If the selected surface is a Non-Sequential Components surface, choosing this option will reverse the incident media and substrate.

Discussion See "POLARIZATION ANALYSIS".

Transmission vs. Wavelength

Computes the S, P, and average polarization intensity coefficients for transmission for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Discussion See "POLARIZATION ANALYSIS".

Absorption vs. Wavelength

Computes the S, P, and average polarization intensity coefficients for absorption for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Diattenuation vs. Wavelength

Computes the R (reflected) and T (transmitted) diattenuation for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Phase vs. Wavelength

For sequential surfaces, this feature computes the S and P polarization phase for reflection (if the glass is MIRROR) or for transmission (if the glass is not MIRROR) as a function of incident wavelength. For non- sequential objects, this feature computes the S and P polarization phase for reflection (if the material is MIRROR) or for transmission (if the material is not MIRROR) as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Retardance vs. Wavelength

Computes the retardance for the specified surface as a function of incident wavelength.

See "Reflectivity vs. Wavelength" for settings

Reports Group

Prescription Data

Reports

Prescription Data

This function generates a list of all surface data, and summarizes the lens system. This is the feature to use for printing the contents of the Lens Data Editor.

General Data   Includes F/#, pupil positions, magnification, etc.

Surface Data   Surface type, radii, thickness, glass, semi-diameter, conic.

Surface Detail   Surface Parameter data. All NSC object data is also listed in this section for NSC groups.

Edge Thickness   The X and Y edge thickness of each surface.

Multi-Config Data   A table of multi-configuration operand data.

Solves/Variables Solve types and data, variables.

Index Data   Index of refraction and TCE data for each wavelength/surface.

Global Vertex   Global coordinates of the vertex and the rotation matrix for each surface. See "Global Coordinate Reference Surface" for a discussion of this data.

Center Of Curvature   Global coordinates of the center of curvature of rotationally symmetric surfaces. The radius, position, and orientation of the surface are used to determine the point at the vertex center of curvature. The location of this point is useful for alignment.

Element Volume Volume in cc, density, and mass. See below.

F/ Numbers   Lists working F/# for each field position and wavelength combination.

Cardinal Points Lists the locations of focal, principal, anti-principal, nodal, and anti-nodal points.

POP Settings   Lists the surface by surface settings used by the Physical Optics Propagation feature.

Files Used   Lists all the files used by the optical system model, including glass data, macros, extensions, CAD files, DLL files, and others.

Discussion This report lists specification data, indices of refraction, global coordinates, element volumes, and more. It is suitable for describing the lens prescription.

Comments on computing element volumes

When Zemax computes element volumes for spherical or plane standard surfaces with circular edges, the edge of the surface with the smaller diameter is assumed to be "squared up" to the semi-diameter of the larger surface. The volume is exact for this common special case.

For other elements, with possibly aspheric faces, elliptical, annular, or rectangular apertures and/or decentered apertures, a numerical integration technique is used that yields approximate volumes to an accuracy of about

0.1%. If the surface is rotationally symmetric, the edge of the surface with the smaller diameter is "squared up" to the semi-diameter of the larger surface. Otherwise, the edge of the surface with the smaller diameter is "tapered" to the semi-diameter of the larger surface. If zero volume is reported than the algorithm is unable to compute the volume to sufficient accuracy. This happens when the aperture types are not the same, or have different sizes or decenters.

When computing the density of elements, the density in grams per cubic centimeter for a catalog glass is retrieved from the glass catalog. For gradient index surfaces, Zemax assumes the density is 3.6 g/cc, which may not be a good estimate.

Universal Plot Group

This feature is only available in the Professional and Premium editions of Zemax.

New Universal Plot 1D

Displays as either a plot or as a text listing the value of any optimization operand as a function of one other parameter. See also "New Universal Plot 2D...".

Independent Variable X Settings

Select either surface, system, multi-configuration, or NSC data as the independent variable.

Select the parameter to use as the independent variable. The available selections include radius, curvature, thickness, conic, parameter, and extra data values if surface parameters are chosen. For system parameters, the selections include the system aperture, field and wavelength data, apodization factor, temperature, and pressure. For configuration data, all multi-configuration operands are listed. For NSC data, object position and parameter data are listed.

The surface number for the independent variable parameter if the parameter is associated with a surface. This field is used for the configuration number if configuration data is selected. If NSC data is selected, this field is used for the object number.

Start/Stop Value The beginning and ending range of the independent variable.

# of Steps The number of values between the start and stop values, inclusive, to compute the dependent variable function value.

Dependent Variable Y Settings

Operand The optimization operand to use as the dependent variable function. The current system merit function is also included as "Merit". If the system merit function is selected, then either the entire merit function or any specific operand's value may be selected using the "Line" control below.

Line The optimization operand # to use as the dependent variable function if "Merit" is selected for the Operand.

Int1/Int2 The Int1 and Int2 values for the selected operand. The dialog box will display the short name for the Int1 and Int2 values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Data1-Data4 The Data values for the selected operand. The dialog box will display the column heading name for these values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Min/Max Plot Value The min and max value to plot for the dependent variable (Y) function. If both of these values are zero, then a default scale will be chosen. If either one is not zero, then the y range of the plot will extend from the minimum to the maximum. If the data does not fit within the specified range, then the data may plot beyond the borders of the data box.

Plot Title Any text may be entered which will be used as the title for the plot or text listing.

Save As New Universal Plot This will save the current settings in a Universal Plot 1D (UPL) file specified by the user. Any name which may be used as a valid file name may be entered. Although the "browse" window allows the file to be saved in any folder, Zemax will only recognize the file if it resides in the <data>\Miscellaneous folder (see "Folders").

Once the settings are saved, the file name will appear as a menu option under the Universal Plot 1D menu option for quick regeneration of the plot.

Load From This will load the settings from a previously saved Universal Plot 1D (UPL) file specified by the drop down list to the right of the button.

Discussion

This feature will either plot as a graph or create a text listing of the value of any optimization operand as a function of a single system, surface, or multi-configuration value.

For example, suppose a plot of the sagittal MTF at 30 lp/mm was needed as a function of the decenter of a lens group (which may be a useful diagnostic for tolerancing analysis). Since the MTFS operand computes the sagittal MTF, the Universal Plot can generate such a graphic or text listing. See "Optimization operands" for a list of available optimization operands. A decenter of a lens group is defined by Parameter 1 or 2 on the relevant coordinate break surface, and Parameter 1 and 2 are both listed in the available surface group parameters.

Because of the number of different plots this feature can generate, there are no intelligent defaults for either the independent or dependent settings. These values need to be carefully provided in the settings dialog box. If the optimization operand cannot be computed, an error message will be displayed and the plot will not be generated.

Because many optimization operands accept Hx and Hy values to define the field point for the calculation, these operands may require that the number of fields be set to 1, then set Hx = 0 and Hy = 1, then finally choose Y Field 1 as the independent variable for a plot of the operand as a function of field. A similar trick works for getting plots as a function of wavelength.

Some individual operands, like DIFF and SUMM, have no meaning if selected from the "Operand" column because these operands are only defined when used as part of a larger merit function. If results from a calculation like DIFF are required, then the merit function should be selected from the Operand list and the individual DIFF's within the merit function selected for the "Line".

When selecting independent variables that are controlled by Multi-Configuration operands, note that the Multi-Configuration operand will override the changes made to the independent variable. For this reason Zemax automatically alters the corresponding data on the Multi-Configuration operand as well.

The escape key will terminate the analysis if the computation is taking too long.

New Universal Plot 2D

Displays as either a plot or as a text listing the value of any optimization operand as a function of two other parameters. See also "New Universal Plot 1D...".

Independent Variable X or Y Settings

Select either surface, system, multi-configuration, or NSC data as the independent variable.

Select the parameter to use as the independent variable. The available selections include radius, curvature, thickness, conic, parameter, and extra data values if surface parameters are chosen. For system parameters, the selections include the system aperture, field and wavelength data, apodization factor, temperature, and pressure. For configuration data, all multi-configuration operands are listed. For NSC data, object position and parameter data are listed.

The surface number for the independent variable parameter if the parameter is associated with a surface. This field is used for the configuration number if configuration data is selected. If NSC data is selected, this field is used for the object number.

Start/Stop Value   The beginning and ending range of the independent variable.

# of Steps   The number of values between the start and stop values, inclusive, to compute the dependent variable function value. The minimum value is 2.

Dependent Variable Settings

Operand The optimization operand to use as the dependent variable function. The current system merit function is also included as "Merit". If the system merit function is selected, then either the entire merit function or any specific operand's value may be selected using the "Line" control below.

Line   The optimization operand # to use as the dependent variable function if "Merit" is selected for the Operand.

Int1/Int2 The Int1 and Int2 values for the selected operand. The dialog box will display the short name for the Int1 and Int2 values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Data1-Data4   The Data values for the selected operand. The dialog box will display the column heading name for these values to aid in identifying the data that needs to be supplied. If the edit control is grayed out; then the selected operand does not use that value.

Display Settings

Show As   Choose surface plot, contour map, grey scale, or false color map as the display option.

Contour Format The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String".

Min/Max Plot Scale   These controls define the min and max range of the Z-direction scale for the display. If the value is set to exactly zero, a default value will be selected. To define a value of zero, use a very small number such as 1.0E-20. If the defined plot scale values do not span the resulting data, the plot may clip, saturate, or otherwise distort the data display. These settings have no affect on the surface or contour plots.

Plot Title   Any text may be entered which will be used as the title for the plot or text listing.

Save As New Universal Plot   This will save the current settings in a Universal Plot 2D (UP2) file specified by the user. Any name which may be used as a valid file name may be entered. Although the "browse" window allows the file to be saved in any folder, Zemax will only recognize the file if it resides in the <data>\Miscellaneous folder (see "Folders").

Once the settings are saved, the file name will appear as a menu option under the Universal Plot menu option for quick regeneration of the plot.

Load From   This will load the settings from a previously saved Universal Plot 2D (UP2) file specified by the drop down list to the right of the button.

Discussion

This feature will either plot as a graph or create a text listing of the value of any optimization operand as a function of any two system, surface, or multi-configuration values.

For example, suppose a plot of the sagittal MTF at 30 lp/mm was needed as a function of the Xand Y decenter of a lens group (which may be a useful diagnostic for tolerancing analysis). Since the MTFS operand computes the sagittal MTF, the Universal Plot 2D can generate such a graphic or text listing. See "Optimization operands" for a list of available optimization operands. The X and Y decenter of a lens group is defined by Parameters 1 and 2 on the relevant coordinate break surface, and Parameters 1 and 2 are both listed in the available surface group parameters.

Because of the number of different plots this feature can generate, there are no intelligent defaults for either the independent or dependent settings. These values need to be carefully provided in the settings dialog box. If the optimization operand cannot be computed, an error message will be displayed and the plot will not be generated.

Because many optimization operands accept Hx and Hy values to define the field point for the calculation, these operands may require that the number of fields be set to 1, then set Hx = 0 and Hy = 1, then finally choose Y Field 1 as

the independent variable for a plot of the operand as a function of field. A similar trick works for getting plots as a function of wavelength.

Some individual operands, like DIFF and SUMM, have no meaning if selected from the "Operand" column because these operands are only defined when used as part of a larger merit function. If results from a calculation like DIFF are required, then the merit function should be selected from the Operand list and the individual DIFF's within the merit function selected for the "Line".

When selecting independent variables that are controlled by Multi-Configuration operands, note that the Multi-Configuration operand will override the changes made to the independent variable. For this reason Zemax automatically alters the corresponding data on the Multi-Configuration operand as well.

The escape key will terminate the analysis if the computation is taking too long.

Applications Group
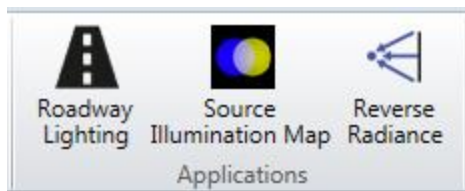
Roadway Lighting

This feature is only available in the Premium edition of Zemax.

Calculates CIE 140-2000 roadway lighting data.
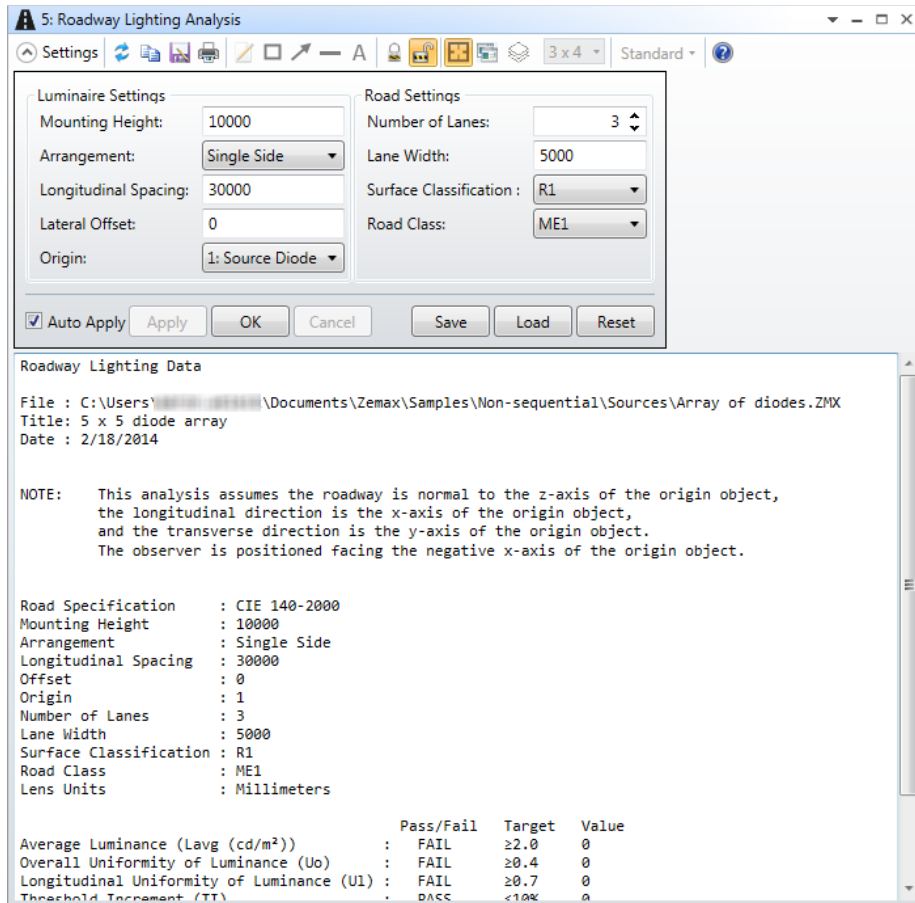
Mounting Height   The distance in lens units from the roadway to the lamp origin.

Arrangement   The arrangement of lamps. Choose single side, double side, or staggered.

Longitudinal Spacing   The spacing between lamps in lens units along the direction of the roadway. For single and double side arrangements, this is the spacing between two adjacent lamps on the same side of the road. For staggered arrangement, this is the spacing along the road of adjacent lamps on opposite sides of the road. The spacing of adjacent lamps on the same side of the roadway would, therefore, be two times the spacing.

Lateral Offset   The lateral lamp offset in lens units. This is the transverse lamp position, relative to the edge of the roadway. This value can be positive or negative.

Origin   The object that represents the origin of the lamp coordinate system.

Number of Lanes   The number of lanes on the roadway.

Lane Width   The width of each lane in the roadway in lens units.

Surface Classification   The road surface classification. This corresponds to the specularity of the road surface. Currently R1, R2, R3, and R4 are supported.

Road Class   The roadway class. This determines the specifications required to have a passing roadway. Currently ME1, ME2, ME3A, ME3B, ME3C, ME3D, ME4A, ME4B, ME5, ME6, CE0, CE1, CE2, CE3, CE4, and CE5 are supported.

Discussion

This analysis computes roadway lighting data according to CIE 140-2000 "Road Lighting Calculations". A quick indication as to whether the lamp passes the specifications dictated by the defined road class is provided along with the

calculated value. While all settings are defined in system units, the computed data is always returned in units of meters and lumens, per the specification.

There are two methods intended for the usage of this analysis. This first is to simply analyze an existing lamp whose far-field intensity is defined in a *.IES or *.LDT file. In this case the far-field data file should be specified as the origin object in the settings and Zemax will only account for that source in the computation. The second is for the user to define sources and geometries associated with the roadway lamp in the system; no other objects or detectors should exist in the system. Zemax will trace analysis rays from all defined sources and compute the data based upon the far-field intensity pattern. In both scenarios the calculation is done assuming the origin object's local x-axis is the longitudinal direction (along the roadway), and the local y-axis is the transverse direction (across the roadway). Note that some IES/LDT files have data defined assuming different lamp orientations. Make sure to confirm this and rotate the source, if necessary, before using this analysis.

An associated capability is the ability to optimize lamps to pass these specifications. See "NSC Roadway Merit

Function Tool" for details.

Source Illumination Map

This feature is only available in the Premium edition of Zemax.

Displays a true color image of the illumination pattern formed by one or more sources. The sources may be defined by IES, LDT, or RSMX source files.
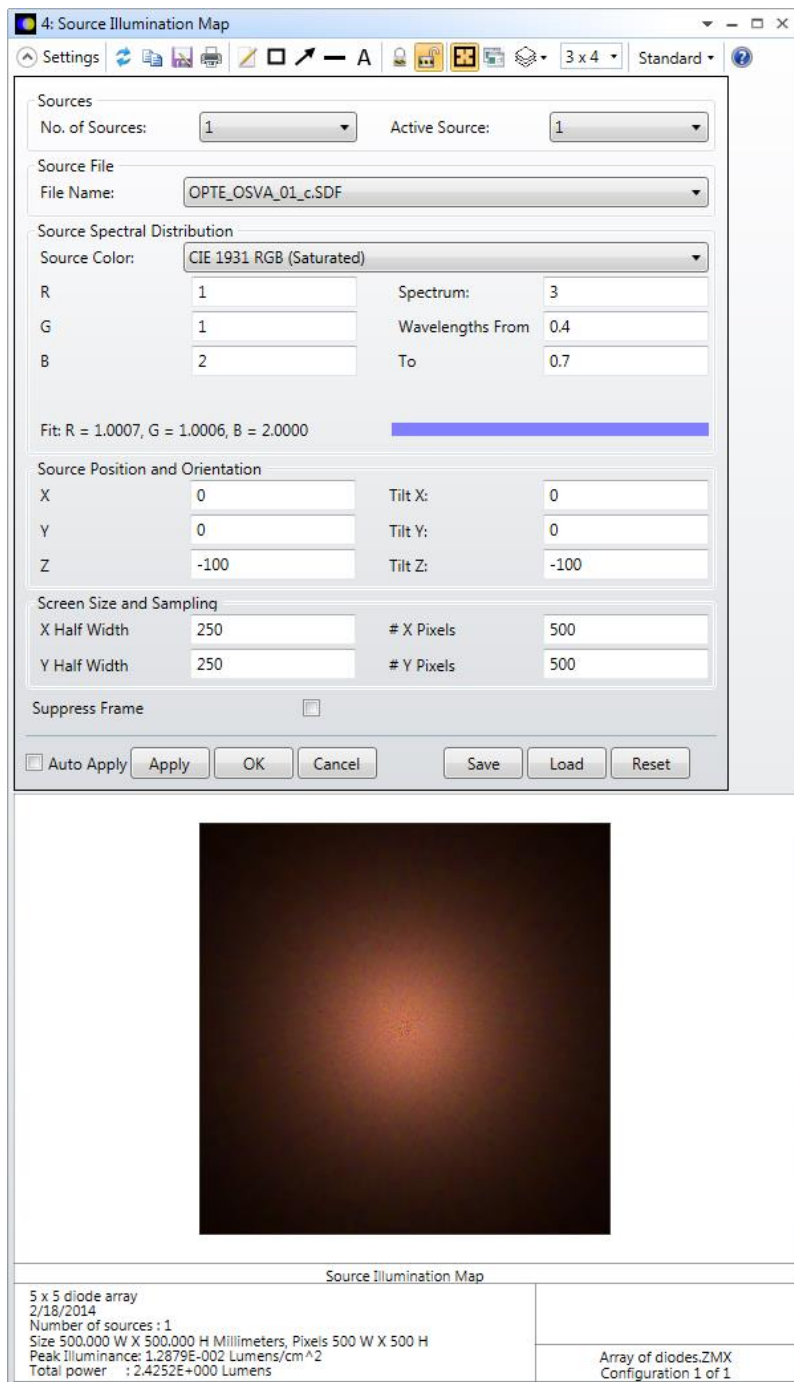
No. of Sources   The number of sources to be displayed. May be any number from 1 to 8.

Active Source   The source number for which the subsequent data (file name, spectral distribution, position and orientation) are defined.

Source File Settings

File Name   The name of the IESNA, EULUMDAT, RSMX, or SDF file containing the data to be displayed.

Source Spectral Distribution Settings

Source Color The model used to define the spectral distribution of the active source. The ten options provided for the source color model are described in "Defining the color and spectral content of sources".

X, Y, Z; x, y; R, G, B; Temp (K); u'v' Six of the models require additional numeric inputs for characterizing the color of the active source. The appropriate inputs should be entered based on the selected model (e.g. X, Y and Z values for the CIE 1931 Tristimulus XYZ model). No additional numeric inputs are required for the System Wavelengths, Uniform Power Spectrum, D65 White, or Spectrum File models. For any model requiring additional numeric inputs - other than Black Body Spectrum - Zemax must perform fitting to determine the corresponding spectrum (this fitting is also required for the D65 White model). More details on the fitting algorithm are provided in "The spectrum fitting algorithm" .

Spectrum The number of wavelengths to use in characterizing the spectral distribution of the active source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

Wavelengths From The minimum wavelength (in microns) to use in characterizing the spectral distribution of the active source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

To The maximum wavelength (in microns) to use in characterizing the spectral distribution of the active source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

Spectrum File The name of the file containing the spectral data for the active source. This option is only available when selecting the Spectrum File model. More information on the file format is provided in "Defining a spectrum file" in the Sources section of the Object Properties.

Source Position and Orientation Settings

X; Y; Z The global X, Y, and Z position of the active source, in lens units. The global position of the screen is always at (0,0,0).

Tilt X; Tilt Y; Tilt Z The global X, Y, and Z tilt of the active source, in degrees.

Screen Size and Sampling

X, Y Half Width The X and Y half width of the screen, in lens units.

 # X, Y Pixels The number of pixels in X and Y on the screen.

Suppress Frame If checked, the frame is not drawn and the entire window is used to display the image.

Discussion

This analysis will simulate the true color image that is formed by luminous flux emanating from a number of sources (up to 8) onto a screen. The far-field luminous intensity data stored in the respective source file, along with the solid angle subtended by each pixel on the screen, determines the luminous flux for each pixel. Since this analysis only considers the far-field distribution of the sources, the spatial source data contained in the RSMX and SDF files are ignored (spatial data are not present in the IES or LDT files).

The total number of sources to simulate is specified by the "No. of Sources" input. The properties of any particular source may then be defined by specifying the source of interest via the "Active Source" input. Each source is then defined by a particular file. In addition, each source may have its own unique spectral distribution and its own unique global position and orientation. If either an RSMX file which contains spectral data or an SDF file is selected for a particular source, then the settings for the source spectral distribution will be grayed out, as in this case the source color information is explicitly provided in either the RSMX or SDF file itself.

The global position of the screen is always at (0,0,0), and the global orientation of the screen is always along the Z axis, i.e. the direction cosines for the normal vector of the screen are (0,0,1). The size and sampling of the screen are given by the "X/Y Half Width" and "# X/Y Pixels" settings, respectively.

Source files must all be stored in the <data>\Objects\Sources\ folders; IES files must be stored in the IESNA sub-folder, LDT files must be stored in the EULUMDAT sub-folder, RSMX files must be stored in the Radiant Source Model™ Files sub-folder, and SDF files must be stored in the Source Files sub-folder (see "Folders").

Reverse Radiance

This feature is only available in the Premium edition of Zemax.

This feature traces rays in an NSC group "backwards" from detector to source, and quickly and efficiently computes the flux incident upon a detector.
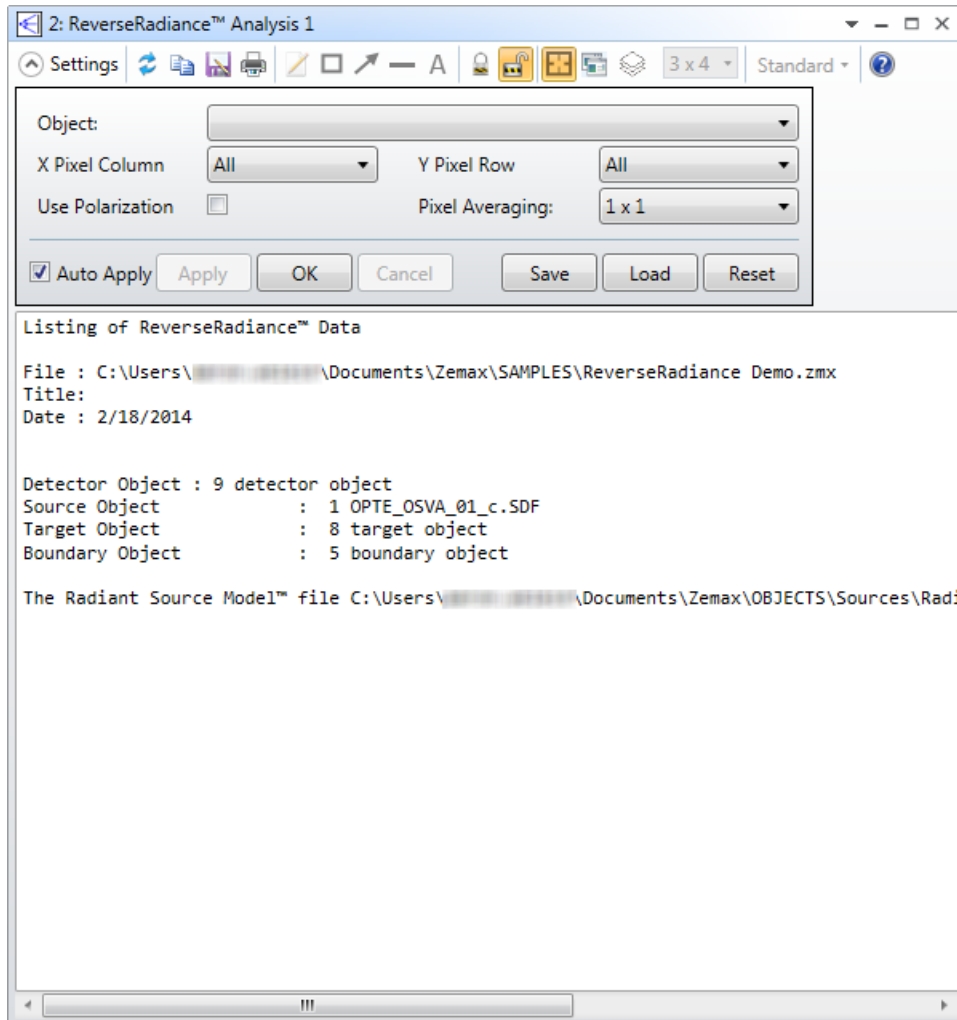
Object   Selects the detector object; see the discussion below for details.

X Pixel Column Selects either all or one specific column.

Y Pixel Row   Selects either all or one specific row.

Use Polarization If checked, polarization transmission is considered in the computation.

Pixel Averaging For sources with very detailed structure in the near field, the luminous intensity of the source can be very sensitive to the exact coordinates of the reverse traced ray. This causes slow convergence and requires high sampling to get good results. Pixel averaging smooths the near field by averaging neighboring pixels when determining

the intensity. For example, a 5 x 5 grid will average the nearest 25 pixels rather than using just the intensity value of the pixel struck.

Discussion

It is a very common analysis task in illumination design to determine the distribution of flux from one or more light sources on a detector. The conventional way of performing this analysis is to describe the flux output from the light source or sources, usually with a list of rays, then trace the rays through all the optical components until they finally hit the detector. The flux from the rays is then summed to determine the flux or flux per area on various pixels on the detector.

Usually, the distribution of both total flux and color varies considerably over the detector. This variation is caused by complexities of the source output, as well as the interaction of the rays with optical components. To resolve and study this variation requires a high number of detector pixels. Unfortunately, the more pixels there are, the smaller each pixel is, and the fewer rays strike the pixel. This results in more error in the computation of how much flux is incident upon each pixel.

The pixel error can be estimated using a Signal-To-Noise Ratio (SNR). If all rays are equally weighted, and all rays are of the same wavelength, then the SNR is reasonably given by the square root of N, where N is the number of rays incident upon the pixel. It is more commonly the case that rays do not all have equal weight, and for true color estimation, many more rays (at the very least, 3 times as many) are required. Therefore, to get results with an accuracy of 1%, requires a SNR of 100, which in turn requires at least 30,000 rays per pixel. For a detector surface with a modest 200 x 200 pixel resolution, the minimum number of rays traced will be on the order of 1 billion rays. When performing conventional foreward direction ray tracing, it is generally not known in advance where the rays will go to (that is what ray tracing reveals), and so an even larger number may be required for high precision results.

There is an additional inefficiency in this approach. Often, the flux is only needed at a few specific points on a reference surface rather than the entire detector. For example, there may be a design goal to have a minimum flux value at one point on a detector, and some other points with either minimum or maximum flux specifications. The forward traced rays may land on regions that are ultimately of no interest in the analysis - these are "wasted" rays whose computations did not contribute to the desired results.

The ReverseRadiance™ Method

There is an alternative method that works very well for some systems. The method involves tracing rays backwards, from the detector to the source. The idea is to trace only those rays that hit a given pixel on the detector. Since it is not know which rays leaving the source hit the detector, the ray is instead traced from the detector towards the source, and the amount of flux is then assigned according to the flux of the source at the position and traveling in that ray direction. There are several complications with this method:

- The ray needs to trace from the detector in a direction that will likely eventually hit the source

- The ray tracing routine must be able to detect that the ray has "hit" the source

- The radiance (or illuminance) of the source must be known given the ray wavelength, position, and angle

- There must be enough sample rays to account for all likely paths to the source from a given detector

The ReverseRadiance™ feature supports this method of computation. There are several steps that must be carefully done for this the analysis to get accurate results, as described below.

Defining the source object

The Source must a Source File as described in "Source File" . In addition, the source file must have been generated using the Radiant Source Model™ ray file generation feature described in "Generate Radiant Source Model Rays" . This step is important, because only the Radiant Source Model™ files contain the actual measured radiant or luminous intensity as a function of wavelength, position, and angle; and this data is required for proper integration of the flux incident upon the detector. Without this data, the ReverseRadiance™ method will compute zero flux on the detector.

Defining the boundary object

Because the source object is a list of rays, it needs a separate object to define the physical extent of the source. This object can be any shape that closely follows the actual physical shape of the source. It may be a sphere, cylinder, rectangular cube, or any other shape supported. It is not required that this object have any physical propertes - its purpose is only to allow the raytrace algorithm to detect when the reverse ray has reached the vicinty of the source. Once the ray is close enough to the source to intersect the boundary object, the ray trace is terminated and the flux is computed as described below.

It is usually a good idea to set the "Rays Ignore Object" to "On Launch" for the boundary object, so that forward tracing rays do not interact with the boundary object. This setting is on the "Type" tab as described in "Type tab" .

Defining the target object

When the reverse rays are launched from the detector, it vastly increases the efficiency and accuracy of the analysis if the rays trace in a direction that ultimately will intersect the boundary object, and by extension, the source itself. The target object is a special type of NSC object called the ReverseRadiance™ Target, as described in "ReverseRadiance Target" .

The target object supports these parameters:

- Shape: Use 0 for a rectangular target, 1 for an elliptical target.

- X/Y Half Width: The size in lens units of the entire target.

- #X/Y (or Radial/Angular) Pixels: This divides the entire target into smaller pixels.

The pixels on the target have a special purpose. One ray will be traced from each detector pixel to the center of each target pixel. This is how the sampling and accuracy of the integration is controlled. More pixels on the target object means a slower computation, but more accurate results. Even with moderately high sampling, far fewer rays are usually needed for accurate results than is the case with forward ray tracing. The number of pixels on the target can be increased until the results converge.

Defining the detector object

Within the NSC Editor, add a ReverseRadiance™ Detector object as defined in "ReverseRadiance Detector". This object defines the position and size of the detector, as well as the number of pixels for which the reverse computation will be performed.

The detector object has several parameters used to define the detector properties:

- Shape: Use 0 for a rectangular detector, 1 for an elliptical detector.

- X/Y Half Width: The size in lens units of the entire detector.

- #X/Y (or Radial/Angular) Pixels: This divides the entire detector into smaller pixels.

- Source Object: This is the object number of the Source File described above.

- Target Object: This is the object number of the ReverseRadiance™ target described above. Boundary Object: This is the object number of the boundary object described above.

- Reverse Ray Color: The pen color to use when drawing rays from this detector. If zero, the default color will be chosen.

Output Data

The feature will generate a text listing of the data computed for each pixel. The Flux is the total power on the pixel, in either watts or lumens. The Irradiance/Illuminance is the power per area. The Flux-X/Y/Z values are the triluminous flux, which give the color of the illuminated pixel. Note the Flux-Y value is redundant with the Flux value in lumens.

Other comments

On layout plots, ReverseRadiance™ rays are traced and drawn from the ReverseRadiance™ Detector to the ReverseRadiance™ target. To preven the layout plot from becoming overly cluttered, the number of target rays is limited to 8 in each direction (X, Y, or Radial, Angular).

When generating ray files using Radiant Source Model™ files, a small file ending in the extension RRD is created along with the Source File SDF file. This file tells the ReverseRadiance™ feature which Radiant Source Model™ file was used

to create the ray set. Both this RRD file, and the original Radiant Source Model™ file, must be present for the ReverseRadiance™ feature to work correctly.

Currently only a single Radiant Source Model™ file is supported. If there are multiple sources present, the

ReverseRadiance™ feature will not be able to account for more than one of the sources at a time.

ReverseRadiance™ will not be accurate for systems containing diffusers (or other scattering components).

The Optimize Tab

This tab provides access to OpticStudio's Optimization Tools.
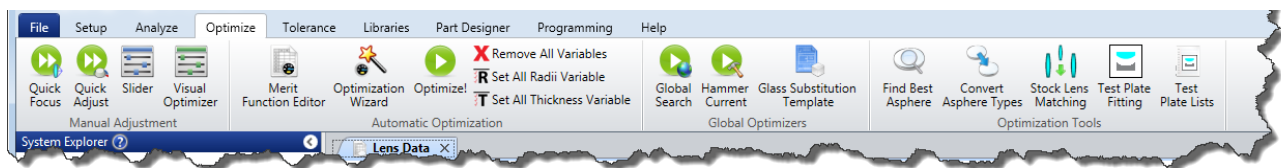
For more information on Optimizing, see Optimization Overview.

Manual Adjustment Group

These are the manual optimization tools.

Quick Focus

The Quick Focus button is available in the Manual Adjustment section of the Optimize tab. The Quick Focus adjusts the back focal distance for best focus. See also "Quick Adjust."
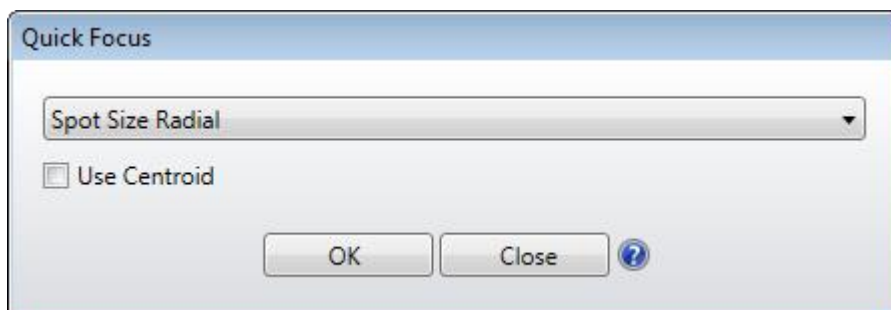
Settings:
Spot Size Radial Focus to best RMS spot radius image surface.

Spot Size X Only Focus to best RMS x direction spot size image surface.

Spot Size Y Only Focus to best RMS y direction spot size image surface.

Wavefront Error Focus to best RMS wavefront error image surface.

Use Centroid Reference all calculations to the image centroid rather than the chief ray. This option is slightly slower, but more appropriate for systems dominated by coma.

Discussion:

This feature adjusts the thickness of the surface prior to the image surface. The thickness chosen is that which minimizes the RMS aberration. There are several different RMS computations that may be performed as listed above. The "best focus" position depends upon which criterion is selected. The RMS is always computed as a polychromatic average over the field, using the defined fields, wavelengths, and weights.

Quick Adjust

The Quick Adjust button is available in the Manual Adjustment section of the Optimize tab. It adjusts any radius or thickness to achieve best transverse or angular ray focus evaluated at any subsequent surface. See also "Quick Focus."
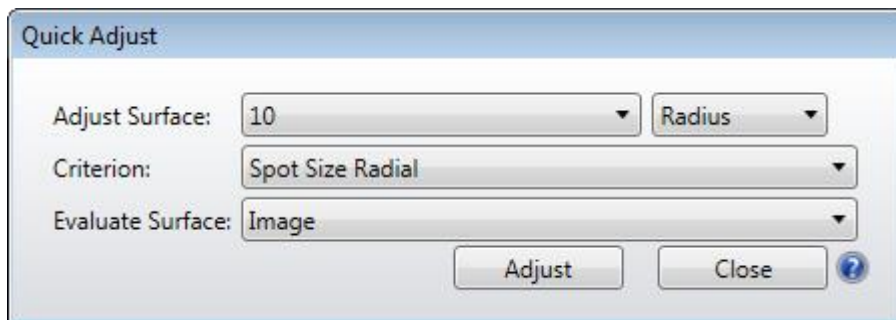
Settings:

Adjust Surface Selects the surface to adjust.

Radius/Thickness Selects either Radius or Thickness to adjust.

Criterion Selects the best focus criterion. All criterion use the image centroid for a reference.

Evaluate Surface Selects the surface at which to evaluate the criterion. Note the criterion is evaluated after refraction into the specified surface for angular criterion.

Discussion:

This feature adjusts any radius or thickness to minimize the RMS aberration at any subsequent surface. The RMS is always computed as a polychromatic average over the field, using the defined fields, wavelengths, and weights. Note RMS angular data is computed after refraction into the specified evaluation surface. This tool can remain open on the desktop while other editing or work is done. Click on the "Adjust" button to recompute the best focus data.

Slider

The Slider is available in the Manual Adjustment section of the Optimize tab. The slider control is used to interactively adjust any system or surface parameter while viewing any analysis window.
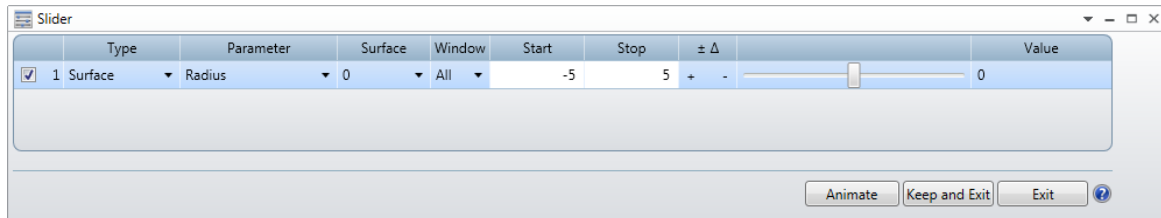
Settings:

Type Select either the surface, system, configuration, or NSC data group.

Parameter Select the parameter to modify. The available selections include radius, curvature, thickness, conic, parameter, and extra data values if surface data are chosen. For system parameters, the selections include the system aperture, field and wavelength data, apodization factor, temperature, and pressure. For configuration data, all multiconfiguration operands are listed. For NSC data, object position and parameter data are listed.

Surface The surface number for the data to be modified if the data is associated with a surface. This field is used for the configuration number if configuration data is selected. If NSC data is selected, this field is used for the object number.

Window Select either "All" or any specific analysis window to update when the slider is adjusted. See the discussion.

Start / Stop Value The beginning and ending range of the data corresponding to the extreme limits of the slider control.

Animate / Stop Automatically increments the data over the defined range, and updates the selected windows in a continuous loop. Press "Stop" (the animate button changes to a "Stop" when animation is running) to terminate the animation.

Keep and Exit Saves the current value in the editor and closes the slider control.

Exit Restores the original data for the modified parameter and closes the slider control.

Discussion:

The slider control can be used to adjust any surface or system parameter while monitoring how the changed value affects the data displayed in any open analysis window or in all windows.

When the slider control is activated, a copy is made of the data to be modified. If a new type of data is selected (for example, the surface number or parameter or surface/system setting is altered) the old data is automatically restored, unless the "Save" button has been pressed. The Save button replaces the saved copy of the original data with the newly modified data.

If any or all of the analysis windows require a long time to update, the slider control may act erratically. This is due to the Windows operating system handling of the slider control requiring a very fast update while dragging or scrolling the slider bar control. If this occurs, select a single window for updating rather than "all" or choose analysis options to reduce computation time.

Visual Optimizer

The Visual Optimizer is available in the Manual Adjustment section of the Optimize tab. This feature uses multiple slider controls to allow simultaneous manual adjustment of selected variable values.
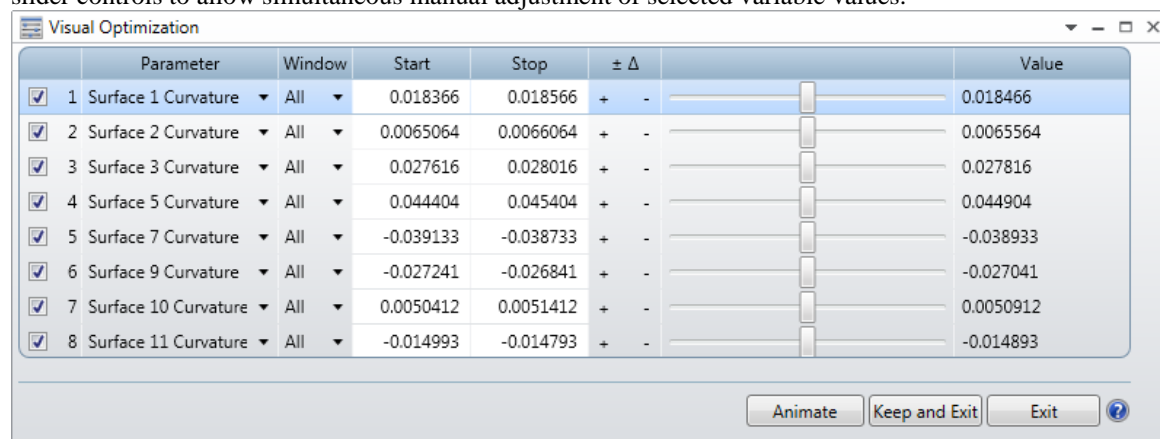
Settings:

Parameter Selects the parameter for that slider to control

Window Select either "All" or any specific analysis window to update when the slider is adjusted. See the discussion.

Start/Stop The minimum and maximum values of the variable, corresponding to the full left and right positions of the slider control, respectively. The difference between the max and min values is the total range of the slider.

Slider The slider control, which may be dragged left or right to adjust the selected variable.

Value The current value of the variable for reference.

Animate / Stop Automatically increments the data over the defined range, and updates the selected windows in a continuous loop. Press "Stop" (the animate button changes to a "Stop" when animation is running) to terminate the animation.

Keep and Exit Saves the current value in the editor and closes the Visual Optimization control.

Exit Restores the original data for the modified parameter and closes the Visual Optimization control.

Discussion:

The visual optimization dialog supports between 1 and 8 individual slider controls. Each slider control may be used to interactively modify any defined variable value. Note the variable status of each value must be set prior to invoking the visual optimization control.

Whenever any variable value is changed, the selected window or all windows are updated in sequential mode. In non-sequential mode, the rays optionally may be traced to update all detector viewers and (optionally) shaded model plots. These operations take time, depending upon the system complexity, the types of analysis windows open, and the number of rays traced. The visual optimization tool will not allow another update to be performed until the current update is completed. It may be required to wait until the updates are completed before the slider may be adjusted again. However, for many illumination systems LightningTrace is fast enough that updates occur almost instantaneously, allowing the performance impact of changing a variable value to be determined in real-time.

Note that radius variables are always varied as curvatures as opposed to radii for numerical stability. The same applies to focal length variables, which are varied as power. This is indicated with the correct labels in the visual optimizer.

Automatic Optimization Group

These are the automatic optimization tools.
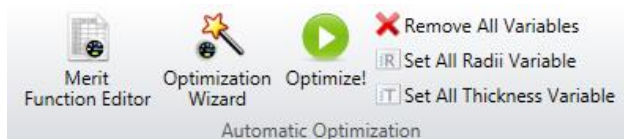
Merit Function Editor

The Merit Function Editor is available in the Automatic Optimization section of the Optimize tab. The Merit Function Editor is used to define, modify, and review the system merit function. The system merit function is used for optimization. For more information, see Optimization Overview.
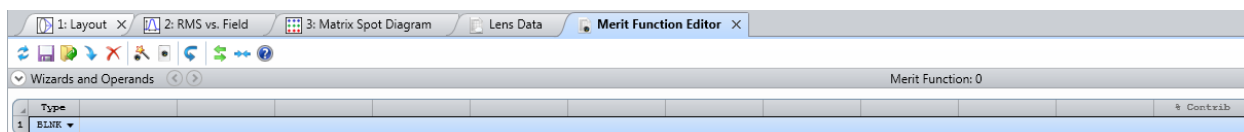
Settings:

Operand Type This option allows the type of operand and other data to be changed. For a complete description of merit function operands, see the following
Optimization Operands by Category
Optimization operands in alphabetical order

Discussion:

Operands are set by typing the name in the first column and then filling in the remaining data fields. There are multiple fields that may be required to define an operand. The fields are called Int1 and Int2 for the two integer values, and Data1 through Data6 for up to 6 double precision values. Not all of the operands use all of the fields provided.

Merit Function Editor Toolbar

The Merit Function Editor Toolbar is available at the top of the Merit Function Editor window. This provides quick access to tools for the Merit Function Editor.
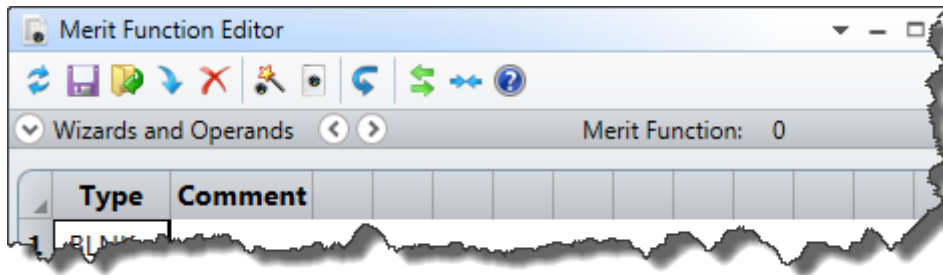
The buttons in the Merit Function Editor Toolbar are listed below.

Update This option recomputes the merit function. All operands are evaluated, and the new values are displayed.

Save Merit Function Saves the current merit function in a *.MF file. This step is only required if the merit function is to be subsequently loaded into another lens. Zemax automatically stores the merit function with the lens when the entire lens is saved.

Load Merit Function Loads a merit function previously stored in a *.MF file or in a *.ZMX file. Either file type may be selected; only the merit function portion of the file will be loaded into the spreadsheet. The current merit function is destroyed.

Insert Merit Function Inserts a merit function previously stored in a *.MF file. Select the row at which to insert the merit function and the file to insert. The existing merit function is preserved.

Delete All Operands Deletes all operands in the merit function.

Optimization Wizard Opens the "Wizards and Operands" tool to set up the merit function with the most common requirements. Can be edited subsequently.

Merit Function Listing Generates a text listing in a separate window of the merit function and shows the split between the user added operands (those operands placed before the DMFS operand) and the default merit function (all operands after the DMFS operand).

Go To Operand Opens a dialog box that allows entry of either the operand number or comment. The editor then displays the desired operand.

Reset Column Order Resets column order to the default settings (see "Editors").

Reset Column Widths Resets column widths to a standard size (see "Editors").

Help Opens the help system.

Optimization Wizard

The Optimization Wizard is available in the Automatic Optimization section of the Optimize tab, or by expanding the bar labeled "Wizards and Operands" above the Merit Function Editor data. The Optimization Wizard displays a dialog box which allows selection of the merit function type and relative ray density.
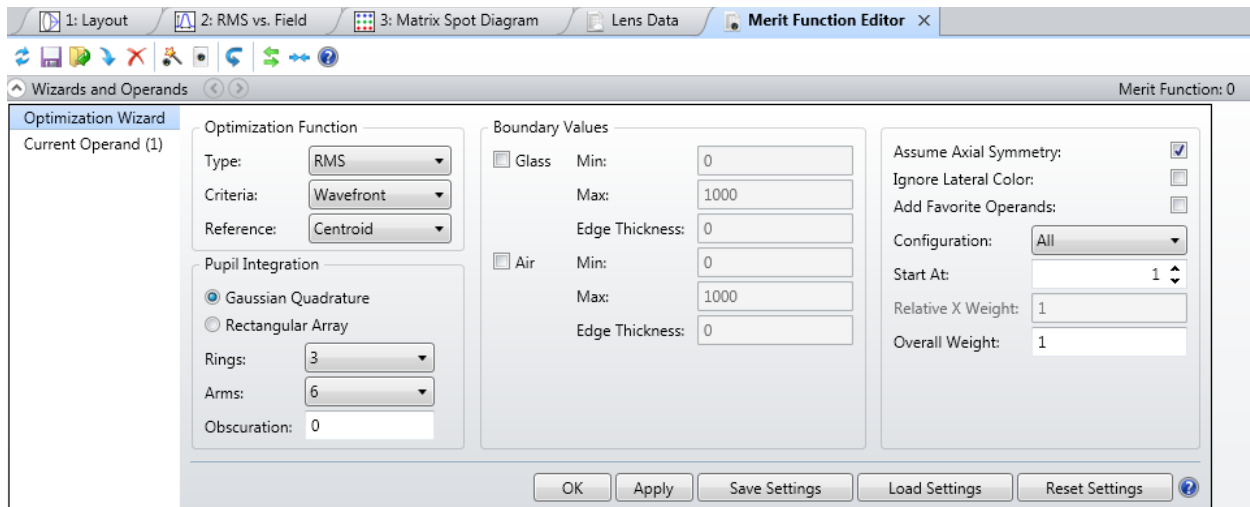
The merit function is a numerical representation of how closely an optical system meets a specified set of goals. Zemax uses a list of operands which individually represent different constraints or goals for the system. Operands represent goals such as image quality, focal length, magnification, and many others.

The merit function is proportional to the square root of the weighted sum of the squares of the difference between the actual and target value of each operand in the list. The merit function is defined this way so a value of zero is ideal. The optimization algorithm will attempt to make the value of this function as small as possible, and so the merit function should be a representation of what you want the system to achieve.

Several different types of merit functions are available. The default merit function is constructed using 4 key choices: The optimization type, criteria, reference point, and integration method. The choices are described in the sections below.

Optimization Function Types

RMS Is an abbreviation for Root-Mean-Square. This type is by far the most commonly used. The RMS is the square root of the average value of the squares of all the individual errors.

PTV Is an abbreviation for Peak-To-Valley. There are rare cases where the RMS is not as important as the maximum extent of the aberrations. For example, if all the rays need to land within a circular region on a detector or fiber. In these cases, the PTV may be a better indicator of performance. This merit function type attempts to minimize the PTV extent of the errors. However, the numerical value of the merit function is not the actual peak-to-valley error.

Physically significant RMS merit functions

Note that the numerical value of the merit function is physically significant when using RMS as the optimization type. If the merit function is RMS-Wavefront-Centroid, then the numerical value of the merit function is the RMS wavefront error in waves. If the merit function is RMS-Spot Radius-Chief, then a value of 0.145 means the RMS spot radius is 0.145 lens units. If the lens units were millimeters, this would correspond to 145 micrometers RMS. If more than one field or wavelength is defined, then the merit function numerical value is the weighted average of the RMS values for the various fields and wavelengths.

Note that optimization using the RMS spot radius merit function will in general yield a different optimum design than the RMS wavefront merit function. The basic reason they are different is that ray aberrations are proportional to the derivative of the wave aberrations. Therefore, it is unreasonable to expect that the minimum of one corresponds to the minimum of the other. A general rule of thumb to use is that if the system is close to diffraction limited (say a PTV wavefront error of less than two waves) then use wavefront error. Otherwise, use spot radius.

Generally speaking, the merit functions which use the centroid as a reference are superior to those that use the chief ray. Most diffraction based performance measures, such as MTF or encircled energy, improve when the RMS wavefront error referenced to the centroid decreases. However, it is always best to re-optimize a final design with the various merit functions to verify which one provides the best performance for the system being designed. For example, the RMS wavefront centroid reference often yields better low frequency MTF response, but worse medial frequency response, than the RMS chief ray reference optimization does.

For PTV merit functions, the merit function numerical value is proportional but not equal to the actual PTV of the aberration. The merit function will however minimize the PTV when the merit function is minimized.

Optimization Function Criteria

Wavefront The aberration measured in waves. See "Comments about RMS wavefront computations."

Spot Radius The radial extent of the transverse ray aberrations in image space.

Spot X The x extent of the transverse ray aberrations in image space.

Spot Y The y extent of the transverse ray aberrations in image space.

Spot X + Y Both the x and y extent of the transverse ray aberrations in image space. The x and y components are considered separately, and both are optimized together. This is similar to Spot Radius, except the signs of the aberrations are retained, which yields better derivatives. Note that in computing the radius of an aberration, the sign information is lost.

Angular Radius The radial extent of the angular aberrations in image space. Intended for afocal systems. See "Afocal Image Space."

Angular X The x extent of the angular ray aberrations in image space. Intended for afocal systems. See "Afocal Image Space."

Angular Y The y extent of the angular ray aberrations in image space. Intended for afocal systems. See "Afocal Image Space."

Optimization Function Reference Points

Centroid The RMS or PTV computation of the data is referenced to the centroid of all the data coming from that field point. Centroid reference is generally preferred, especially for wavefront optimization. For wavefront optimization, reference to the centroid subtracts out the mean wavefront over the entire pupil, x-tilt, and y-tilt, none of which degrade image quality. Centroid reference also yields more meaningful results when coma is present, since coma shifts the image centroid away from the chief ray location.

Chief The RMS or PTV computation of the data is referenced to the chief ray at the primary wavelength. For wavefront optimization, reference to the chief ray subtracts out the mean wavefront over the entire pupil, but not x-tilt or y-tilt. Note the exact point at which the OPD is defined to be zero is arbitrary; this is the reason the chief ray reference subtracts out the mean wavefront.

Unreferenced This option is only available if the data is wavefront. If the wavefront is unreferenced, then the OPD data with respect to the chief ray is used without subtracting the mean wavefront or tilt.

Pupil Integration Settings

There are two different pupil integration methods used to construct the merit function: Gaussian quadrature (GQ) or rectangular array (RA). The GQ algorithm is superior for almost all cases of practical interest.

Gaussian Quadrature

The GQ algorithm uses a carefully selected and weighted ray set to accurately compute the RMS or PTV error over a uniformly illuminated entrance pupil (strictly speaking, the PTV algorithm is not a GQ algorithm, but it is very similar). The weighting for all rays is applied according to the weights set on the wavelength and field in the System Explorer and by the GQ merit function algorithm. For RMS merit functions, the weighting and ray set selection used is based on a method described in a paper by G. W. Forbes, JOSA A 5, p1943, 1988. For the PTV merit functions, the ray set is based on solutions to the Chebyshev polynomials, described in Numerical Recipes, Cambridge University Press (1989). If you are interested in detailed information on the basis and accuracy of these methods, see these references. GQ is much more accurate than any other known method, and requires fewer rays. Therefore, you get the best of both worlds: greater speed and greater accuracy.

The only drawback to GQ is that the algorithm assumes the pupil is a circle, or more generally, an ellipse. For non-elliptical pupils, GQ does not work accurately. For example, if there are surface apertures in the optical system that vignette enough rays to alter the effective shape of the pupil significantly, GQ should not be used.

One notable exception is when using circular pupils with central obscurations, such as a Newtonian telescope. In this case, an input value may be given to indicate the fraction of the pupil that is obscured (the "Obscuration"). The method used to account for obscuration of the circular pupil within the GQ algorithm is described in B.J. Bauman and H. Xiao, "Gaussian quadrature for optical design with non-circular pupil and fields, and broad wavelengths", Proc. SPIE, Vol. 7652, p76522S-1 (2010). While, in general, modest central obscurations do not affect the RMS significantly because the aberrations tend to be smaller in the central zone of the pupil, the use of a finite "Obscuration" factor within the GQ algorithm allows for a fully accurate representation of the obscuration in the merit function calculation.

Note also that GQ works fine when used with vignetting factors, since the ray pattern is redistributed from a circle to an ellipse. More details on the use of vignetting factors in conjunction with GQ may be found in the Knowledge Base article entitled "How to use Vignetting Factors".

The GQ algorithm requires specification of the number of "Rings", the number of "Arms", and the "Obscuration" factor.

Rings The "Rings" setting determines how many rays are traced at each field and at each wavelength. For on-axis fields (zero degrees field angle in a rotationally symmetric system), the number of rays is equal to the number of rings. For all other fields in symmetric systems, the number of rays traced per ring is equal to half the number of "arms" (defined in the next paragraph). Only half the rays are traced because the left-right symmetry of the system is exploited. Each set of rays is traced for each defined wavelength. For example, if you have one on-axis field, two off-axis fields, three wavelengths, four rings, and six arms selected, the number of rays traced is 3 * (4 + 4*3 + 4*3) = 84. For systems without rotational symmetry, the number of rays per ring is the number of arms independent of field. In the prior example, this means 3 * 3 * 4 * 6 = 216 rays. Zemax automatically calculates these numbers for you; the only reason it is described here is so you will understand how the default merit function is defined. Optimization runs are longer if more rays are traced.

Arms The "Arms" setting determines how many radial arms of rays in the pupil are traced. By default six equally spaced (in angle) arms are traced (or three if the system is rotationally symmetric). This number may be changed to eight, ten, or twelve. For most common optical systems, six is sufficient.

Sampling considerations for Rings and Arms

You should select the number of rings and the number of arms according to the order of aberrations present in your system. The highest order of aberration the GQ algorithm can accurately integrate is (2*n - 1) where n is the number of rings. If an optical design will be limited by 5th order ray aberrations n should be at least 3. A simple way of determining the correct number of rings is to select the minimum number, one. Then go to the optimization dialog box and note the merit function. Now go back to the default merit function tool, and select two rings. If the merit function changes by more than a few percent, go back and select three, and so on until the merit function does not change significantly (perhaps 1%). Repeat the procedure for the number of arms (six arms is almost always plenty). Selecting more rings or arms than required will not improve the optimization performance; it will only slow the algorithm down needlessly. Tracing more rays than required will not help you find better solutions!

> **Selecting more rings or arms than required will not improve the optimization performance; it will only slow the algorithm down needlessly.**

Obscuration The "Obscuration" setting determines the fraction of the circular pupil that is vignetted, i.e. the fraction of the pupil through which rays cannot be traced. The default value is 0, i.e. the default assumes no obscuration. The maximum possible value is 1, which would indicate a fully-obscured pupil through which no rays could be traced. To avoid a situation in which no rays are defined in the merit function, Zemax will reset the input value to 0 if the user sets the (absolute value of the) input value to be >= 1.

Rectangular Array

The RA algorithm traces a grid of rays through the pupil. The advantage to the RA algorithm is the ability to accurately account for the effects of vignetting in the merit function. This is useful in systems such as obscured telescopes and camera lenses which intentionally clip troublesome rays. The disadvantage to the RA algorithm is speed and accuracy. Usually, more rays are required to achieve a given degree of accuracy than the GQ algorithm. The bottom line: don't use RA unless you are using surface apertures.

The RA algorithm requires specification of the "Grid" size, and the "Delete Vignetted" option.

Grid Determines the number of rays to be used. The grid size can be 4x4 (16 rays per field per wavelength), 6x6 (36 rays per field per wavelength) etc. Rays on the grid are automatically omitted if they fall outside the entrance pupil, so the actual number of rays used will be lower than the grid size squared. Selecting a larger grid size generally yields more accurate results at the expense of slower execution. However, there may be an advantage in choosing a large grid density, and then selecting the "Delete Vignetted" checkbox (described in the next paragraph). The reason is that a large grid density will fill the pupil with rays, and then the operands which are vignetted will be deleted. The result is a reasonable number of rays which accurately reflect the aperture of the system.

Delete Vignetted If selected, each ray in the merit function will be traced through the system, but if it is vignetted by a surface aperture, if it misses any surface, or if it is total internal reflected at any surface, then the ray is deleted from the merit function. This keeps the total number of rays in the merit function to a minimum. The disadvantage is that if the vignetting changes as the design is optimized, then the merit function may have to be regenerated. It is always a better choice to use the vignetting factors and then use the GQ algorithm than to delete vignetted rays if possible. Vignetting factors can be adjusted, if required, during optimization using SVIG in the merit function.

Note that Zemax will attempt to trace any ray defined in the merit function regardless if that ray is vignetted or not. For example, if the chief ray height is targeted using REAY, and there is a central obscuration that vignettes the chief ray, Zemax will still trace the ray and use the operand results as long as the ray can be traced. Zemax does not check to see if defined rays are vignetted, because this introduces substantial overhead during optimization.

In general, avoid vignetting of rays by surface apertures, and use vignetting factors to shape the beam size when possible. To optimize on the fraction of unvignetted rays, a macro must be defined to perform the required computations. However, this method is very prone to stagnation during optimization because small changes in lens parameters lead to discrete changes in the merit function as rays abruptly jump from being vignetted to not being vignetted.

Boundary Settings

Boundary constraints may be automatically generated and included in the default merit function by checking the air and/or glass boundary values on.

Glass If selected, then MNCG, MXCG, and MNEG operands will be added to the merit function to constrain the minimum center thickness, maximum center thickness, and minimum edge thickness for glass surfaces, respectively.

Air MNCA, MXCA, and MNEA operands will be added to the merit function to constrain the minimum center thickness, maximum center thickness, and minimum edge thickness for air spaces, respectively.

The automatic boundary constraint feature is meant to save some manual entry of routine boundary constraints on optical systems with or without mirrors. More complex lenses, such as those with complex coordinate breaks, or multi-configurations usually require additional boundary constraints to be added to the merit function manually.

Other Settings

Assume Axial Symmetry If selected, then the default merit function will exploit the left-right and rotational symmetry of the lens when constructing and evaluating the merit function. Fewer rays will be traced, accelerating the optimization with no loss of accuracy. In systems with coordinate breaks or non-rotationally symmetric systems, the default is unselected, which means symmetry will not be exploited. Overriding the default system symmetry is useful, however, if you are designing a lens that Zemax thinks is non-symmetric, but the lack of symmetry does not affect the aberrations. For example, if tilted but flat fold mirrors are present, these mirrors do not eliminate the left-right symmetry of the system, but Zemax will by default assume symmetry does not exist. Some gradient index surfaces also use non-symmetric index variation terms which are often zero (they are used only for tolerancing).

Select the checkbox to accelerate the optimization in these cases. See also the "USYM" operand description. If the merit function will be used for tolerancing, check this option off, because even symmetric lenses generally become non-symmetric during tolerancing.

Ignore Lateral Color By default, Zemax references all RMS or PTV computations to a common reference point for each field. All of the rays are traced for all wavelengths for each field point, and the primary wavelength chief ray or the centroid of all the rays is used as the reference point. If "Ignore Lateral color" is selected, then Zemax computes an independent reference point for each wavelength as well. This is useful for designing systems that intentionally divide the beam by wavelength, such as a prism or spectrometer system. This option will cause the merit function to optimize each color spot independently.

Add Favorite Operands When checked, Zemax will add the merit function defined in FAVORITE.MF above the DMFS line in the default merit function. This merit function file must be placed in the <data>\MeritFunction folder (see "Folders").

Configuration Select one or all configurations for the merit function.

Start At This option is used to add the default merit function at a specific position within the Merit Function Editor operand list. To control what the default start at value will be, see the DFMS operand definition. Use a value of 0 to start at the end of the current merit function.

Relative X Weight The relative X weight is the additional weighting to be placed on the X component of the transverse aberrations when computing the PTV or RMS SPOT X + Y merit function. This setting has no effect on the other merit functions. If the relative X weight is less than unity, then the Y components are weighted more heavily; if the relative weight is greater than unity, then the X components are weighted more heavily. If left at the default value of unity; then

---

the components are equally weighted. This control is useful for systems which intentionally form slit images, such as spectrometers.

Overall Weight When switching between different default merit function types, such as RMS spot radius and RMS wavefront error, the numerical magnitude of the default merit function can change dramatically. This may make it tedious to manually adjust the weights of operands that are not part of the default merit function. The overall weight is a factor that scales all the weights in the default merit function. Under most circumstances, this weight may be left at one.

Button Functions

OK Closes the Optimization Wizard tool and adds the generated merit function to the exiting merit function at the specified starting line.

Apply Adds the generated merit function to the exiting merit function at the specified starting line.

Save Settings Saves the settings in the Optimization Wizard dialogue box. After the settings are changed or reset, the saved settings are re-loaded by clicking the "Load Settings" button.

Load Settings Loads the Optimization Wizard settings that were last saved.

Reset Settings Resets settings to default.

Optimize!

The "Optimize!" button is available in the Automatic Optimization section of the Optimize tab. This button displays the Local Optimization tool for automatic optimization of the current optical system.
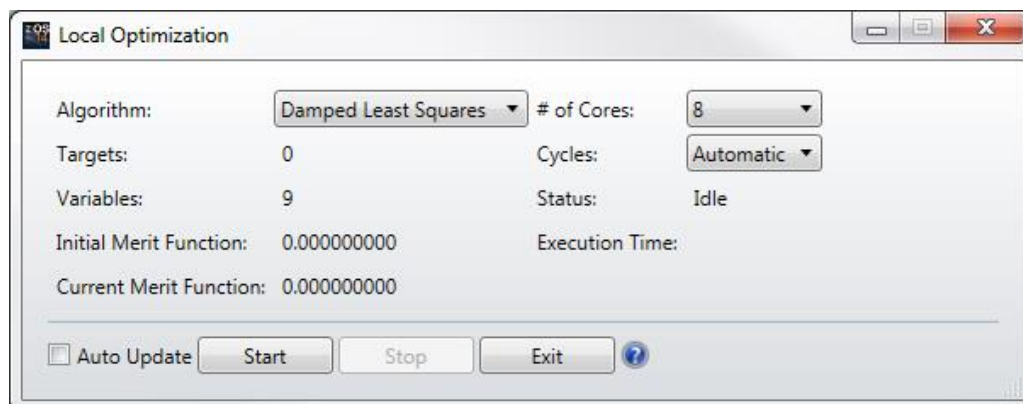
Settings:

Algorithm Local Optimization uses a Damped Least-Squares or Orthogonal Descent algorithm to improve or modify a design to meet specific conditions.

Damped Least Squares (DLS) uses numerically computed derivatives to determine a direction in solution space which produces a design with a lower merit function. This gradient method has been developed specifically for optical system design and is recommended for all optimization problems.

Orthogonal Descent (OD) uses an orthonormalization of the variables and discrete sampling of solution space to reduce the merit function. The OD algorithm does not compute numerical derivatives of the merit function. For systems with

inherently noisy merit functions, such as non-sequential systems, OD will usually outperform DLS. For a discussion of a similar algorithm, see S. Kudaev, P. Schreiber, "Automated optimization of non-imaging optics for luminaries", in Optical Design and Engineering II; Laurent Mazuray, Rolf Wartmann, Eds., Proc. SPIE 5962, p. 87-95 (2005).

Note that both DLS and OD are local optimization algorithms whose final results depend upon the starting point. For global optimization, see "Global Optimizers".

# of Cores Selects the number of cores over which to spread the optimization task. More than 1 may be selected, even on a single CPU computer, in which case the single CPU will time share the multiple simultaneous tasks. The default is the number of processors detected by the operating system.

Cycles Select the number of optimization cycles. 1 Cycle Executes a single optimization cycle, 5 Cycles Executes 5 optimization cycles, etc. Inf. Cycles Executes optimization cycles in an infinite, continuous loop until "Stop" is pressed.

Selecting automatic will cause the optimizer to run until no progress is being made. Automatic mode is highly recommended. The time required to run a given optimization cycle varies enormously with the number of variables, the complexity of the system, the number of solves, the number of operands, and of course the computer speed. If the cycle is taking too long, or if it appears to be hung up, or if you feel the design is not progressing adequately, click on Stop to end the optimization run.

Auto Update If checked, Zemax will automatically update and redraw all open windows at the end of each optimization cycle. This allows monitoring of the optimization progress using any analysis feature.

Start Starts the optimization.

Stop Stops a running optimization, and returns control back to the dialog box.

Exit Closes the Local Optimization dialog box.

Discussion:

The Local Optimization (or simply, "Optimization") feature provided by Zemax is quite powerful, and is capable of improving lens designs given a reasonable starting point and a set of variable parameters. Variables can be curvatures, thicknesses, glasses, conics, parameter data, extra data, and any of the numeric multi-configuration data. Zemax uses either an actively damped least squares or an orthogonal descent algorithm. The algorithms are capable of optimizing a merit function composed of weighted target values; these target values are called "operands". Zemax has several different default merit functions which can be changed easily using the Merit Function Editor. For details on this procedure see "Modifying the merit function."

When the optimization begins, Zemax first updates the system merit function. If any of the operands cannot be computed, the optimization cannot begin, and an error message will be displayed. Operands cannot be computed if they require the tracing of rays which miss surfaces or which undergo total internal reflection (TIR) at an index boundary. If such an error message appears, usually the starting lens prescription is in error, or the ray targets are incorrectly defined (this will not happen with the default merit functions, but might happen with user defined rays). Zemax can automatically recover if the merit function cannot be evaluated during the course of optimization; only the starting system need be adequate to compute all operands in the merit function.

See "Optimization Overview" for a full discussion of building merit functions, setting variables, and using the optimization feature.

Remove All Variables

The Remove All Variables button is available in the Automatic Optimization section of the Optimize tab. This button provides a quick way to remove all of the variables defined in the current lens.
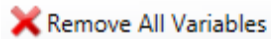
Discussion:
Variables are removed by setting each of the values to "fixed" at the current value.

Set All Radii Variable

The Set All Radii Variable button is available in the Automatic Optimization section of the Optimize tab. This button provides a quick way to make all radii variable.
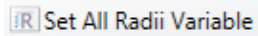
Discussion:
Variables are only set on curved surfaces that define the boundary between two different media. A variable radius is designated with a "V" listed next to the Radius column in the Lens Data window.

Set All Thickness Variable

The Set All Thickness Variable button is available in the Automatic Optimization section of the Optimize tab. This button provides a quick way to make all thicknesses variable.

Discussion:
Variables are only set on finite non-zero thicknesses. A variable thickness is designated with a "V" listed next to the Thickness column in the Lens Data window.

Global Optimizers Group

These are the global optimization tools.

For more information on Global Optimization, see Global Optimization.

Global Optimizer

The Global Optimizer is available in the Global Optimizers section of the Optimize tab. This feature initiates a search for the global optimum, which is the best possible design for a given merit function and set of variables.
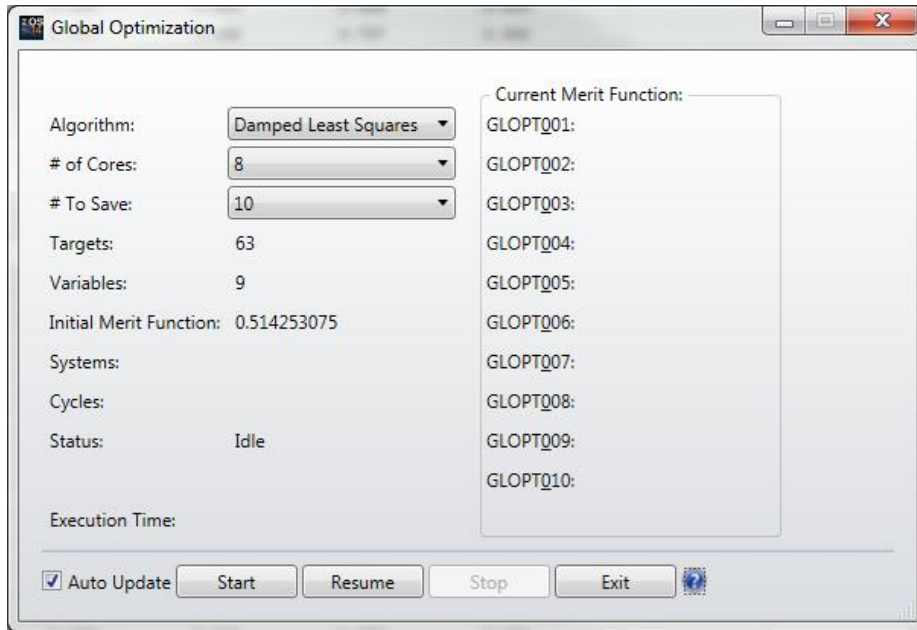
For more information see "The Global Optimization algorithm" in the "Optimization Overview" reference file.

Settings:

Algorithm The algorithm control allows selection of either Damped Least Squares (DLS) or Orthogonal Descent (OD). The DLS algorithm is preferred for most imaging systems, while OD is preferred for systems with noisy, low precision merit functions, such as illumination systems. For more information, see "Performing an optimization".

# of Cores Selects the number of cores over which to spread the optimization task. More than 1 may be selected, even on a single CPU computer, in which case the single CPU will time share the multiple simultaneous tasks. The default is the number of processors detected by the operating system.

 # To Save Selects the number of lenses to save. Zemax copies the initial file into new Zemax format files called GLOPT_001.ZMX through GLOPT_nnn.ZMX where nnn is the maximum number of files to save. Zemax will then begin looking at various combinations of lens parameters extracted from the ranges you have defined. The optimization will proceed on a newly generated lens until Zemax concludes the new lens has been sufficiently optimized.

As each new lens is generated, Zemax will compare the merit function of the new lens to the best lenses found so far, and will place it in the correct location in the best lenses list, renaming other lens files as required. If the lens has a higher merit function than all of the lenses on the best list, then it is discarded. The cycle repeats indefinitely. Each time a new lens is found which is better than the worst in the list of best lenses, it is placed in the correct place in the list. After hundreds of lenses have been replaced (which may require many tens of thousands of lenses to be evaluated) the resulting set will hopefully contain some very good designs, or at least some promising forms.

The global optimization dialog box will also display the merit functions of the ten best lenses found so far. If the number of lenses to save is greater than ten, these files are stored on disk, but the merit functions are not displayed.

The algorithm also periodically returns to lenses in the best list to see if they can be improved upon. Occasionally some lenses will be improved and placed back in the list. If this happens, the older design being replaced is rejected if it has the same basic form as the new lens. This is done to keep some diversity in the best list, otherwise all the lenses will be of nearly identical form.

Auto Update If checked, all open windows will be updated every time a lower merit function is found.

Start Starts the optimization.

Resume Is very similar to the Start button, however, Resume will first load the existing GLOPT_xxx files and place their current merit functions in the best list. Thus, Resume begins the search from where a previous run ended. Resume does not erase the existing best files, whereas Start erases the files and begins a new search entirely based upon the lens currently in the Lens Data Editor. If Resume is selected when a completely unrelated file is in the Lens Data Editor, Zemax will attempt to optimize the lens using the current data in the Lens Data and Merit Function Editors, but will use the old GLOPT files respective merit functions for comparison purposes.

Stop Stops a running optimization, and returns control back to the dialog box. Depending upon what the algorithm is doing, it may exit immediately, or it may require several seconds. Once the algorithm has terminated, you can click on Exit. You can now open any of the GLOPT_xxx.ZMX files for further analysis.

Exit Closes the Global Optimization dialog box.

Hammer Optimizer

The Hammer Optimizer is available in the Global Optimizers section of the Optimize tab. This feature automates the repetitive optimization of a design to escape local minima in the merit function.
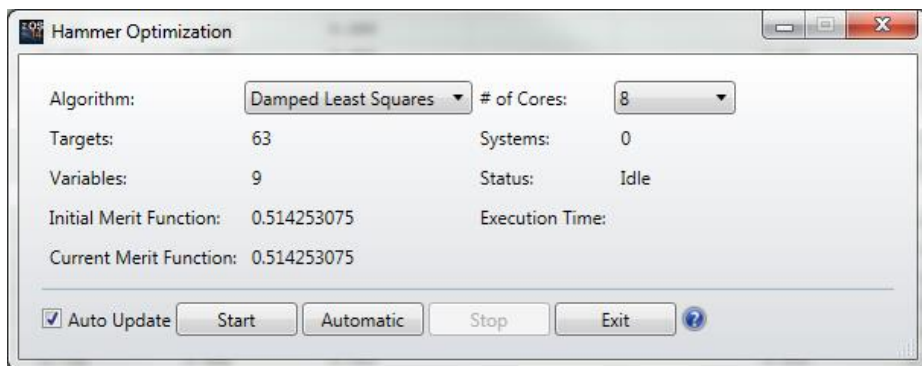
For more information on the "Hammer Optimization algorithm" in the "Optimization Overview" reference file.

Settings:

Algorithm Choose either Damped Least Squares (DLS) or Orthogonal Descent (OD). The DLS algorithm is preferred for most imaging systems, while OD is preferred for systems with noisy, low-precision merit functions, such as illumination systems. For more information, see "Performing an optimization".

# of Cores Selects the number of cores over which to spread the optimization task. More than 1 may be selected, even on a single CPU computer, in which case the single CPU will time share the multiple simultaneous tasks. The default is the number of processors detected by the operating system.

Auto Update If checked, all open windows will be updated every time a lower merit function is found.

Start Begins the Hammer Optimization loop. Hammer Optimization will take the lens and exhaustively attempt to refine it by making adjustments and optimizations. Each time the lens is improved, it will be saved to disk in a temporary file.

Automatic Calls the conventional (local) damped least squares (DLS) or orthogonal descent (OD) optimizer and runs it in Automatic mode. Sometimes it is useful to optimize a lens before calling Hammer Optimization if the lens has not already been optimized.

Stop Terminates the Hammer Optimization.

Exit Closes the dialog box.

The Hammer Optimization screen shows the starting merit function, and the best merit function found so far. Although good results can occasionally be had in several minutes, the algorithm should be allowed to run for several hours, and preferably overnight. To terminate the search, select Stop, then Exit.

If Zemax abnormally terminates, the last saved file can be found in the temporary file. The temporary file name is constructed from the starting lens file name. If the lens being optimized is stored in the file "C:\Zemax\SAMPLES\MYFILE.ZMX" then the temporary file will be called "C:\Zemax\SAMPLES\MYFILE_HAMMER.ZMX".

Glass Substitution Template

The Glass Substitution Template is available in the Global Optimizers section of the Optimize tab. This feature limits the range of acceptable glasses to choose based on cost and other constraints when using the Hammer and Global Optimizers.
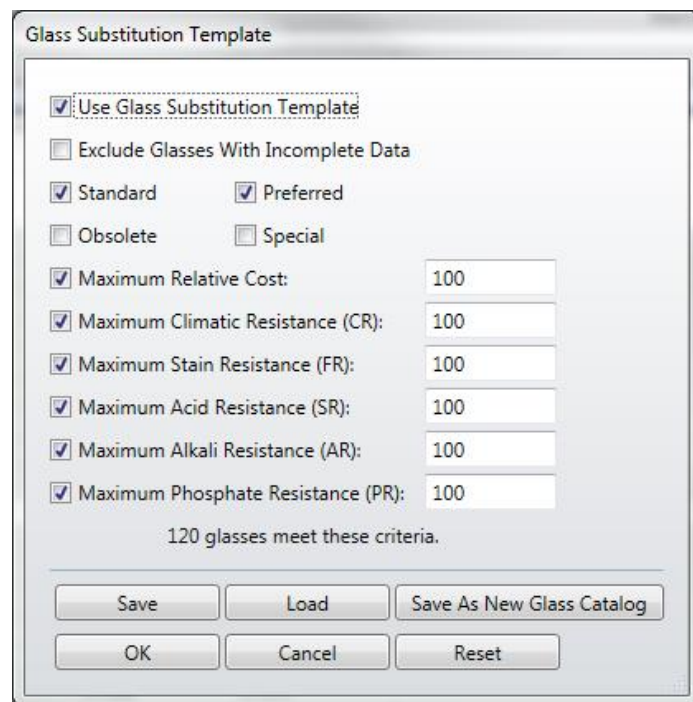
Settings:

Use Glass Substitution Template If checked, the defined parameters are used to restrict the glasses available for substitution by the global optimizer.

Exclude Glasses With Incomplete Data If checked, glasses with incomplete template data will not be selected by the global optimizer. Only data corresponding to non-zero template values are tested before excluding a glass.

Standard, Preferred, Obsolete, Special These checkboxes indicate the acceptable status settings of glasses. These options are additive. For example, if both Standard and Obsolete are checked, then either Standard or Obsolete glasses are acceptable, Note a single glass may have only one status.

Maximum Relative Cost The maximum relative cost. The cost of Schott BK7 is 1.0, all other glasses are more expensive, and have larger relative costs. Use zero to ignore relative cost, or uncheck the box next to this item.

Maximum Climatic Resistance (CR) The maximum allowed value of the CR code. Use zero to ignore CR. Use zero to ignore relative cost, or uncheck the box next to this item.

Maximum Stain Resistance (FR) The maximum allowed value of the FR code. Use zero to ignore FR, or uncheck the box next to this item.

Maximum Acid Resistance (SR) The maximum allowed value of the SR code. Use zero to ignore SR, or uncheck the box next to this item.

Maximum Alkali Resistance (AR) The maximum allowed value of the AR code. Use zero to ignore AR, or uncheck the box next to this item.

Maximum Phosphate Resistance (PR) The maximum allowed value of the PR code. Use zero to ignore PR, or uncheck the box next to this item.

Save Saves the current settings to a *.GST file.

Load Loads the current settings from a previously saved *.GST file.

Save As New Glass Catalog Saves all the glasses that meets the current criteria into a new glass catalog.

OK Accepts the current settings.

Cancel Reverts to the prior settings.

Reset Restores the default settings.

Discussion:

If "Use Glass Substitution Template" is unchecked, any glass defined in the current catalogs may be selected if not explicitly excluded (by having "exclude substitution" checked in the glass catalog) and if the wavelength range is suitable for the current lens. For glasses that do not include the cost or resistance codes, a "?" will be displayed in the glass catalog. If no cost or resistance data is provided for a glass, the glass will not be selected if "Exclude Glasses With No Data" is checked. Template data is stored along with the lens file; so the template parameters are lens file specific. If no glass meets the specified template parameters, the glass substitution status is automatically removed.

The glass substitution template is also used when converting from model to real glasses. When converting model glasses to real glasses, it is possible that no glass can be found that meets the template specification. If this happens, Zemax will first ignore the "Exclude Glasses With No Data" selection and then try again to find a suitable glass. If still no glass can be found that meets the criteria, then Zemax ignores the template entirely and tries again. If still no suitable glass can be found, glass number 1 in the catalog, whatever properties it may have, will be used, even if the glass does not have a suitable wavelength range or is far from the model glass parameters.

Individual tests for Relative Cost, CR, FR, SR, AR, and PR may be activated and deactivated by checking the box next to each description.

Optimization Tools Group

These are the optimization tools.

Find Best Asphere

The Find Best Asphere tool is available in the Optimization Tools section of the Optimize tab. This feature replaces spherical surfaces with aspherical surfaces and reoptimizes the design to determine which surface would benefit the most from aspherization.
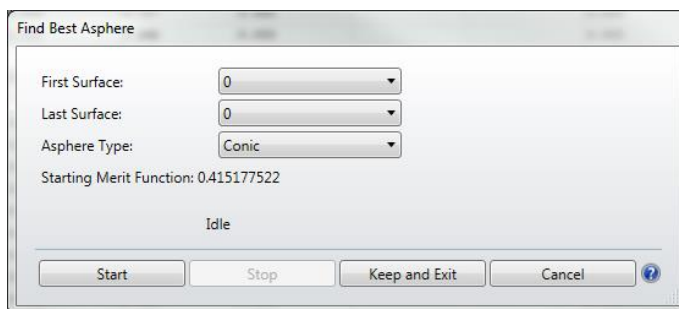
Settings:

First/Last Surface These controls set the range of surfaces to consider adding an asphere to.

Asphere Type Choose an asphere defined by a conic or an even polynomial asphere to the specified maximum order.

Start Begins the analysis. After an analysis is complete or stopped, selecting Start again will discard the previous results and search again for the best asphere using the current surface range and asphere type selections.

Stop Terminates a running analysis.

Keep and Exit Retains the best asphere found and exits.

Cancel Discards the changes made and exits.

Discussion:

This feature automates the process of determining which spherical surface, if re-optimized to be an aspheric surface, would provide the lowest merit function. Each surface is evaluated to see if it is a candidate asphere. To be considered, the surface must be of type Standard, have no conic value, define a boundary between air and glass (cemented surfaces usually make poor aspheres), have a curvature that is either variable or controlled by a marginal ray angle or F/# solve,

and have a curvature that is not under control of the Multi-configuration Editor. Surfaces that do not meet this test are ignored.

When a candidate surface is identified, the surface is converted into an asphere of the user-selected type. The aspheric terms are set as variables for optimization. The local damped-least squares optimizer is then called to optimize the modified system. If the resulting system has the lowest merit function yet found, the system is retained. The procedure repeats until all surfaces have been tested. Finally, this feature reports which surface, when converted to an asphere, provided the lowest merit function. Note the currently defined merit function is used, and all data that are variable are re-optimized during this process. The current merit function should be appropriate for an aspheric design, which may require higher sampling than a non-aspheric design for good optimization.

Note that like all local optimization results, there is no way to know if the solution found is the optimum "global minimum" for that combination of merit function, variables, and design parameters. For this reason, once the best candidate asphere is determined, it is usually a good idea to run the Hammer Optimization on the resulting design to see if any further gains are possible.

No attempt is made to determine whether the resulting asphere is practical to fabricate, or is more or less costly to manufacture as compared to making other surfaces aspheric. Some care must be taken in using this feature, as the results may not be useful for multi-configuration systems, especially those using ignored surfaces in some configurations but not in others. To prevent any particular surface from being selected as a possible asphere, either remove the variable status from the radius of curvature, or temporarily enter a small conic value (such as 1E-20). This feature will then not include that surface in the possible candidate aspheres.

Convert Asphere Types

The Convert Asphere Types tool is available in the Optimization Tools section of the Optimize tab. This feature converts an aspheric surface from one type of asphere to another, and computes the coefficients that best match the sag of the existing surface type. The even, extended, and Q-type (Forbes) aspheres are supported.
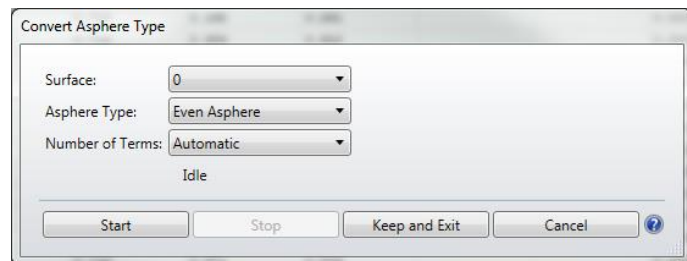
Settings:

Surface Choose the surface to convert.

Asphere Type Choose one of the supported asphere surface models to convert to.

Number of Terms The number of aspheric terms to use in the conversion, or automatic to allow Zemax to make a reasonable guess.

Start Begins the conversion process.

Stop Terminates a running conversion.

Keep and Exit Retains the converted asphere and exits.

Cancel Discards the converted asphere and exits.

Discussion:

This feature automates the process of converting from one type of asphere to another. Currently the even, extended, and Q-type aspheres are supported. Zemax automatically determines the proper order for the new asphere based upon the order of the old asphere, or a specific number of terms may be selected. The conversion is done by a least squares fitting so that the new asphere provides the minimum RMS difference in the surface sag between the old and new aspheres. Only the semi-diameter of the surface is used in determining the region over which the fit is done, and for aspheres that use the normalization radius, the semi-diameter is used for this normalization radius. If the radius or conic of the starting asphere are either non-zero or variable, then these parameters are also made variable in the fitting process and the new asphere may modify these values. The residual RMS surface sag error in lens units is reported at the end of the fitting process. If the asphere conversion is valid, this should be a very small number relative to the semi diameter of the surface. If the residual is too large, a higher setting for the number of terms may be used to reduce the residual error.

Stock Lens Matching

The Stock Lens Matching tool is available in the Optimization Tools section of the Optimize tab. This tool finds stock lens matches for lenses defined in the current system.
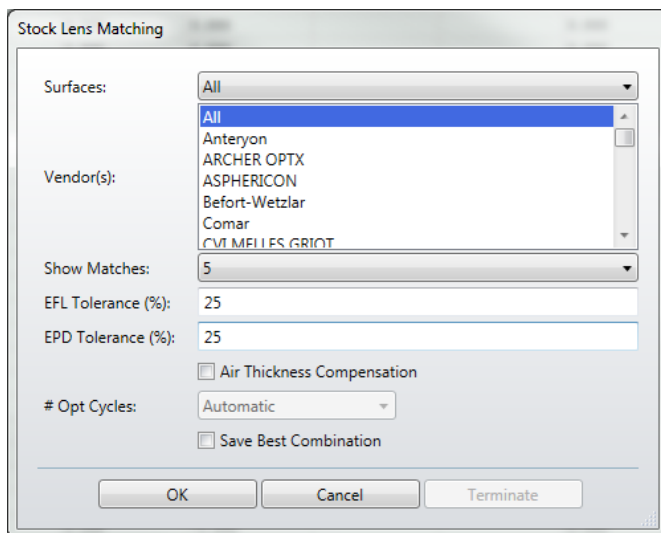
Settings:

Surfaces The lens surfaces to match. Choose All or Variables. If All is chosen, all glass elements with glass surfaces are matched. If Variables is chosen, all glass elements that have variable radius status are matched.

Vendor(s) Choose all or specific stock lens vendors.

Show Matches The maximum number of stock lens matches to show.

EFL Tolerance (%) The maximum EFL tolerance that matched stock lenses can deviate from the defined lens. This value is the EFL difference divided by the EFL of the defined lens expressed as a percent.

EPD Tolerance (%) The maximum EPD tolerance that matched stock lenses can deviate from the diameter of the defined lens. The EPD of matched lenses will not be smaller than the diameter of the defined lens. This value is the EPD difference divided by the diameter of the defined lens expressed as a percent.

Air Thickness Compensation If selected, air thicknesses allocated as variables will be reoptimized after the stock lens is in place using the specified number of damped least squares (DLS) cycles.

 # Opt. Cycles The number of cycles of DLS to use when optimizing air thicknesses. This option is only available when air thickness compensation is enabled.

Save Best Combination If selected, the best combination of lenses will be saved in the current directory with the name {lens_name}_SLM.

Discussion:

This feature automates the process of finding suitable stock lens replacements for as designed lenses in an optical system. Zemax will search through the selected vendor catalogs for matches for the selected lens elements. The EFL tolerance setting and EPD tolerance setting are used to limit the stock lenses considered. Additional limiting criteria defined internally are shape factor, wavelength range, and number of elements. When Zemax finds a potential match, the stock lens data is copied into the original system and the merit function is evaluated. A list of the top matches for each component and the resulting merit function are displayed in the text listing results. In addition, the best combinations of the top matches are evaluated to find the best combination of stock lenses. These results are also listed in the text results.

This feature is intended to be used with spherical singlets and cemented, multi-element lenses. Air-spaced doublets and other multi-element lenses will be matched as individual elements. Aspheric, GRIN, and toroidal surfaces are not supported. The matching process can be time-consuming depending on the number of elements to match, the number of vendors selected, and the complexity of the merit function. A progress indicator is shown in the title bar of the dialog to give an indication of the matching progress, though it is not possible to pre-determine the exact time it will take to complete the matching. If the process is taking too long, the terminate button may be used to abort the process.

Note that many stock lenses use encrypted coatings provided with the default COATING.DAT file. If a different coating file is used, it is necessary to copy over the stock lens encrypted coatings found at the end of COATING.DAT. Zemax will issue a warning if a coating file other than COATING.DAT is used with this tool. This tool will only work with newer, binary format lens catalog (ZMF) files; older ZMF files may be in text format. All current vendor catalogs and catalogs created with the Make Private Lens Catalog tool are in binary format.

Test Plate Fitting

The Test Plate Fitting tool is available in the Optimization Tools section of the Optimize tab. This tool provides automatic fitting of radii to a vendor test plate list. Only radii which are variable prior to executing this tool are fit.
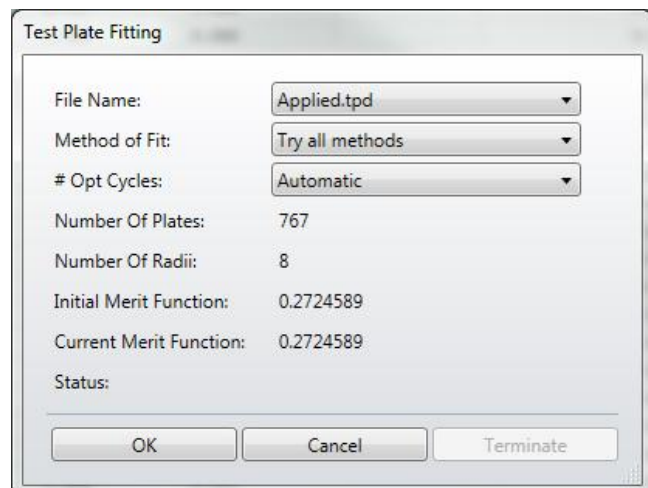
Settings:

File Name Selects test plate list.

Method of Fit Selects the fitting order, which may affect the quality of the fit. See below.

 # Opt Cycles Selects the number of optimization cycles to run, or Automatic.

Discussion:

The current merit function is used as a figure of merit during the fitting process. To match a particular radius, make the radii variable on the lens data, or make the CRVT operand corresponding to the radii variable on the multi-configuration editor. As many radii as desired can be matched at one time. Zemax will attempt to fit all radii with variable status, even if the radii correspond to surfaces that are not spheres, such as toroids, cylinders, or other aspheric surfaces. To prevent fitting of the radii on these surfaces, remove the variable flag before performing the fitting.

Now select the test plate fitting tool. Select which vendor test plate file is to be used. The fitting method may be selected as follows:

> Try All Methods: Try all of the following methods, and uses whichever one yields the lowest merit function.

> Best To Worst: Fits the radii with the closest test plate match (measured in fringes) first.

> Worst To Best: Fits the worst fitting radii first.

> Long To Short: Fits the longest radii first.

> Short To Long: Fits the shortest radii first.

Press OK to begin the fitting. Before any fitting is performed, Zemax will do a local optimization on the starting system. If it is not fully optimized, the starting radii may differ from what is shown in the lens data editor. Zemax then proceeds to search the test plate list for the closest match (in fringes) between all of the radii and all of the test plates. Each test plate must be of the correct shape (convex or concave or both, as required) and must have sufficient diameter to test the clear aperture of the lens surface (as determined by the semi-diameter value on the Lens Data Editor). The test plate has sufficient diameter if the diameter is at least 3/4 of the clear aperture diameter of the lens surface (if the test plate diameter is defined as zero, this test is not applied and any semi-diameter value is accepted). The surface being fitted must also have an edge sag no more than 0.2 mm different than the test plate at the semi-diameter of the surface. In practice this is hundreds of fringes of power and test plates that do not meet this test would likely increase the merit function unacceptably.

The radius of the test plate that best fits one of the radii is then substituted in for the actual radius. The variability of the chosen radius is removed, and the lens is reoptimized. For this reason it is important to allow compensators such as spacings to be variable as well as the radii to be fit. The reoptimization will adjust all remaining variables, including the remaining radii. Note the optimization will use the current merit function. After optimization, if there are any variable radii remaining, the procedure repeats. Note that the radii are not fit in general in the order they appear in the lens data or multi-configuration editors.

During the fitting process, the number of radii remaining and the current merit function will be presented. After all of the radii have been fit, a report will be presented. The report will provide the vendor identification information, and a list of the radii that were changed and which vendor test plate ID numbers were selected.

There is no way to know if the test plates selected are optimal. If there are a large number of test plates in the test plate list, and the values are reasonably continuous without large gaps, the fitting will usually be quite good.

If the merit function increases unacceptably during the fitting process, either a different vendor test plate list should be used, the design needs to be modified, or some of the lenses may have to have custom test plates made.

Usually, the last radii to be fit are the likely candidates for custom test plates. The report shows the order in which the radii were fit.

Using the "Try All Methods" option will yield the lowest overall merit function, since all 4 methods are tried and the only the one yielding the lowest merit function is retained. However, there may be some algorithm other than the 4 listed that might yield a better fit.

There are occasions where the algorithm is unable to find any suitable test plate match for a given radius. This can occur when no test plate with a large enough diameter is sufficiently close to the desired radii. If this occurs, a message stating "NO MATCH FOUND!" will be printed in the report file, and that radii will subsequently be ignored. Usually this means test plates for this radii will have to be custom made.

All test plate data is provided by the respective vendors, and no warranty is provided as to the accuracy or completeness of the data. For details on the latest vendor test plate lists, contact the vendor. New test plate files may be added to Zemax. The test plate data files end in the extension .TPD, and must be placed in the <data>\Testplate folder (see "Folders"). The text files use the following format:

```
 ! Header line 1
 ! Header line 2
 ! etc... up to 15 lines of header info is supported
 partname radius diameter code
 partname radius diameter code
 partname radius diameter code
 partname radius diameter code
 etc...
```

where partname is an ID number or name for the test plate, radius is the radius of curvature in millimeters, diameter is the actual plate diameter in millimeters, and code is an integer value. The code should be -1 for concave only, 0 for both concave and convex, and 1 for convex only. All four values must be on one line, separated by spaces or tabs. No spaces or tabs are allowed in the partname parameter. Each test plate (or pair) should be on one line, and the lines separated by carriage returns. The numeric values can be in free format, but must be in millimeters. The maximum number of test plates allowed per file is 30,000. The header lines can be used for any purpose, and typically indicate the vendor's name, address, phone, e-mail, and other contact data. The "!" is replaced by a space when Zemax lists the data out on test plate lists and fitting reports.

Test Plate Lists

The Test Plate Lists tool is available in the Optimization Tools section of the Optimize tab. This tool displays in a text window a list of test plates from a specific vendor.
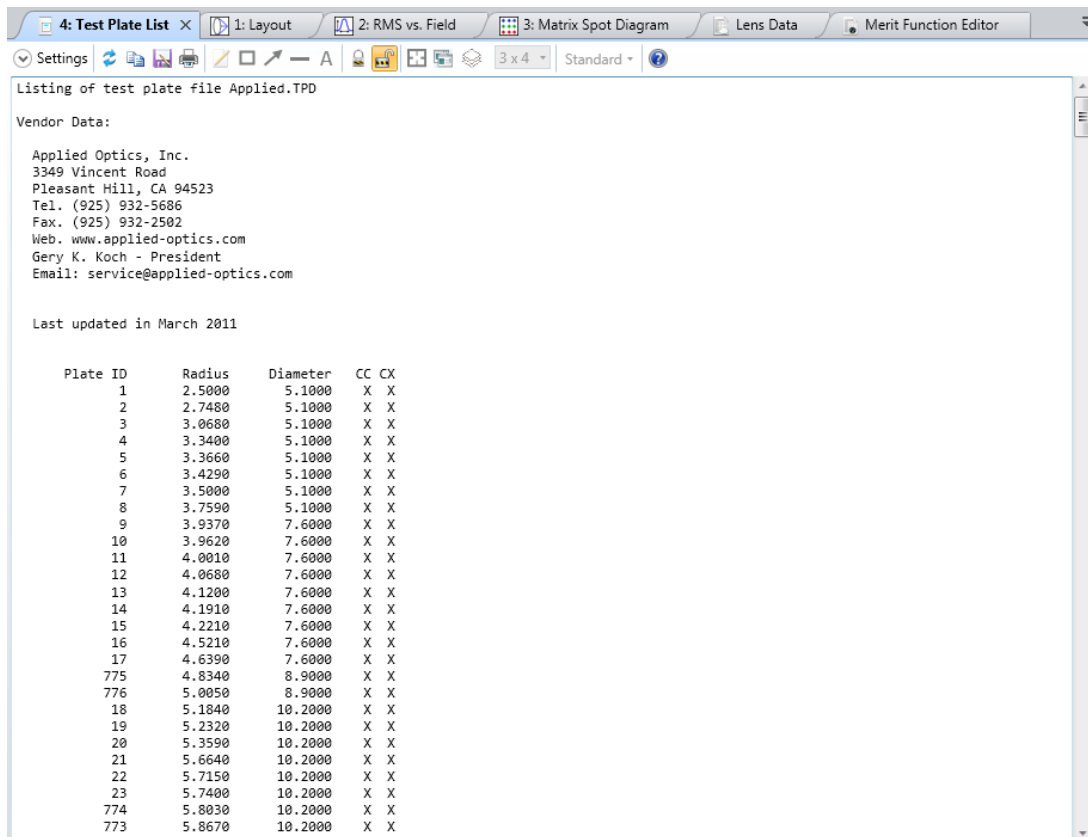
Settings:

File Name To change the name of the test plate file, expand the settings in the upper left corner of the window. The file must be in the <data>\Testplate folder (see "Folders")
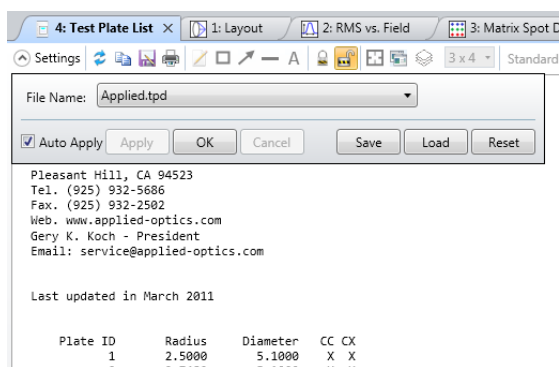
Discussion:

All units on the report are in mm. The CC and CX columns indicate the availability of concave and convex test plates, respectively.

The Optimize Tab (Non-Sequential UI Mode)

This tab provides access to OpticStudio's Optimization Tools. In Non-Sequential UI Mode, the Optimize Tab has a more limited number of tools than in Sequential UI Mode.
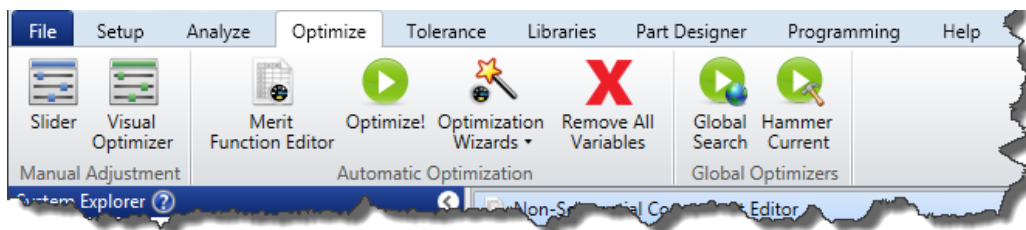
Manual Adjustment Group (NSC)

These are the manual optimization tools available in Non-Sequential UI Mode.
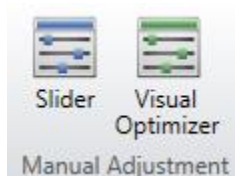
Slider (NSC)

The Slider is available in the Manual Adjustment section of the Optimize tab. The slider control is used to interactively adjust any system or surface parameter while viewing any analysis window.
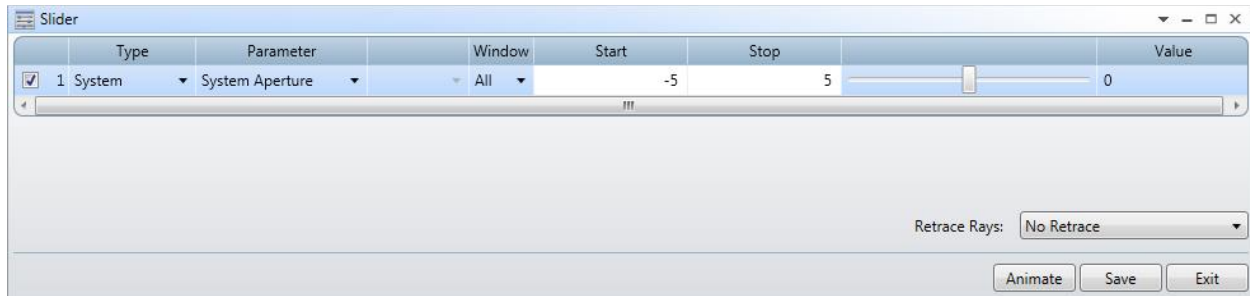
See the description of "Slider" under the "Manual Adjustment" section of "The Optimize Tab".

In non-sequential mode, there also exists an option to Retrace Rays.

Settings:

No Retrace Will not retrace rays.

Use Rays Conventional ray tracing is performed using the total number of analysis rays defined for all sources to update all detector viewers and (optionally) shaded model plots.

Use LightningTrace (with all 6 levels of sampling supported for LightningTrace) The LightningTrace analysis is performed to update all detector viewers and (optionally) shaded model plots. For ray sampling less than 64X, edge sampling of 4X is used, otherwise edge sampling of 16X is used. See "LightningTrace Control" of "The Analyze Tab".

Visual Optimizer (NSC)

The Visual Optimizer is available in the Manual Adjustment section of the Optimize tab. This feature uses multiple slider controls to allow simultaneous manual adjustment of selected variable values.
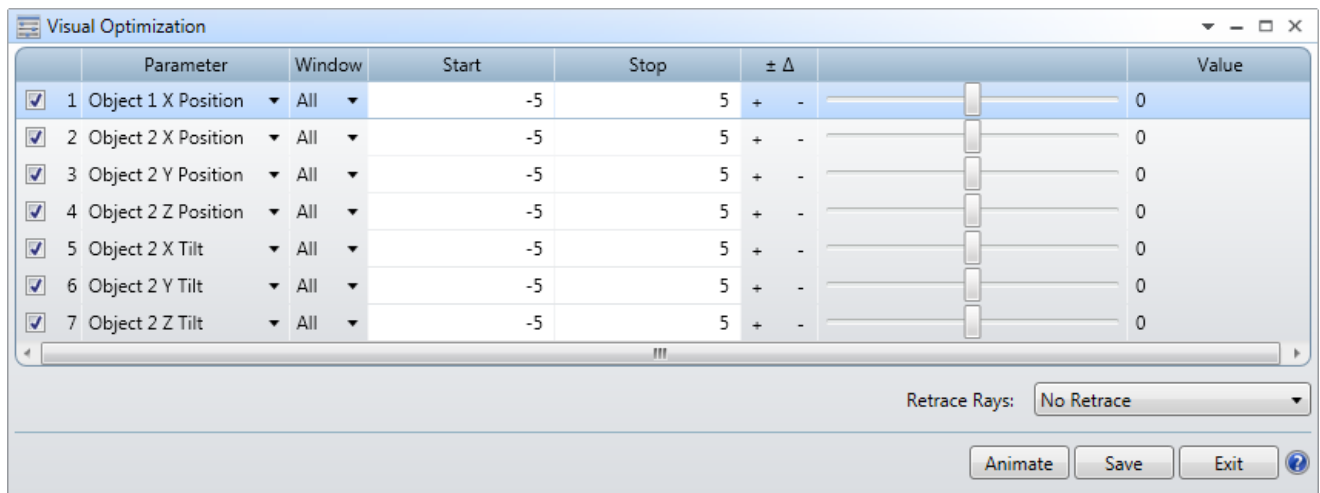
See the description of "Visual Optimizer" under the "Manual Adjustment" section of "The Optimize Tab".

In non-sequential mode, there also exists an option to Retrace Rays.

Settings:

No Retrace Will not retrace rays.

Use Rays Conventional ray tracing is performed using the total number of analysis rays defined for all sources to update all detector viewers and (optionally) shaded model plots.

Use LightningTrace (with all 6 levels of sampling supported for LightningTrace) The LightningTrace analysis is performed to update all detector viewers and (optionally) shaded model plots. For ray sampling less than 64X, edge sampling of 4X is used, otherwise edge sampling of 16X is used. See "LightningTrace Control" of "The Analyze Tab".

Automatic Optimization Group (NSC)

These are the automatic optimization tools available in Non-Sequential UI Mode.
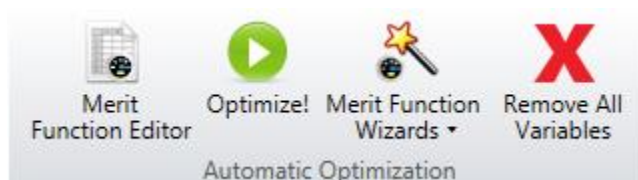
Merit Function Editor (NSC)

See the description of "Merit Function Editor" under the "Automatic Optimization" section of "The Optimize Tab".

NSC Operands

Optimizing with sources and detectors in non-sequential mode

Optimizing illumination systems or other optical systems that use non-sequential sources and detectors is supported using the NSDC, NSDD, NSDE, NSDP and NSTR operands.

A typical merit function would consist of three groups of operands:

First, **NSDD** operands would be used to clear the data in the current detectors. Use NSDD with the detector number set to zero to clear all energy in all detectors. Usually a single NSDD at the top of the merit function is all that is needed. NSDD returns a value of zero and has no effect on the merit function value when used to clear detectors.

Second, **NSTR** operands are used to trace rays from NSC sources. NSTR i traces analysis rays from source i; NSTR 0 traces all analysis rays from all sources. Note the number of analysis rays on the NSC editor determines how many rays are traced and how long the evaluation of the NSTR operand will take. NSTR always returns a value of zero and has no effect on the merit function value. The NSTR operand supports options to split, scatter, and use polarization.

Third, a new group of **NSDC, NSDD, NSDE,** or **NSDP** operands are used to read out the detector data. NSDD has four arguments: surface, detector, pixel, and data. Surface is the surface number of the NSC group (use 1 if the program mode is non-sequential). Detector is the object number of the detector. Both detector objects and faceted detectors may be used as detectors. If pixel is an integer greater than zero, the flux, flux/area, or flux/solid angle is returned for that pixel. Which of the three is determined by the data argument, which should be 0, 1, or 2 for flux, irradiance, or intensity, respectively. If pixel is 0, the sum of all the flux or flux/area in the detector is returned. The units of the returned data is determined by the system units, see "Analysis Units". NSDD can compute other data. For a full description of the pixel argument, see "NSDD". If the object is a faceted detector, or if the pixel number is -1 or 0, only data options 0 and 1 are supported. Similar capability to that described for NSDD exists for coherent data using NSDC, for detector color data using NSDE, and for detector polar data using NSDP.

The practical difficulty in optimizing these systems is the difficulty of computing derivatives of detected energy with respect to variable parameters because of the relatively large uncertainty in computing detected energy. Many rays must be traced to determine illumination patterns approximately.

Comments about random numbers and NSTR

Note that when launching rays with the NSTR operand, Zemax seeds the random number generator with the identical value every time the merit function is evaluated. This means the identical set of random numbers are used on every evaluation of the merit function. If no changes are made to the optical system, the merit function will always evaluate to a consistent value; a desirable property for the optimization algorithm to function. However, if a change is made to the system, rays may take different paths and the identical set of random numbers may be used in different ways; for example, in the computation of scattering paths.

Pixel numbering for detectors

For a detailed discussion of the pixel numbering used by the various types of detectors, see "NSC - Detectors".

Optimize! (NSC)

See the description of "Optimize!" under the "Automatic Optimization" section of "The Optimize Tab".

Merit Function Wizards (NSC)

These are the Merit Function Wizards, available in the Automatic Optimization section of the Optimize Tab in Non-Sequential UI Mode.
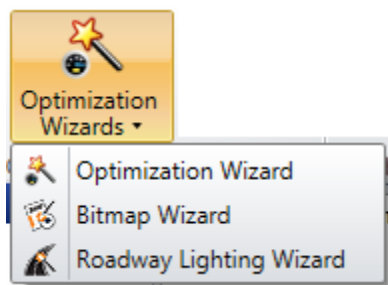
The Wizards can also be accessed by expanding the "Wizards and Operands" bar above the Merit Function Editor data.

Note that wizards may not be available if the specified objects are not loaded in the NSC Editor, as shown below.
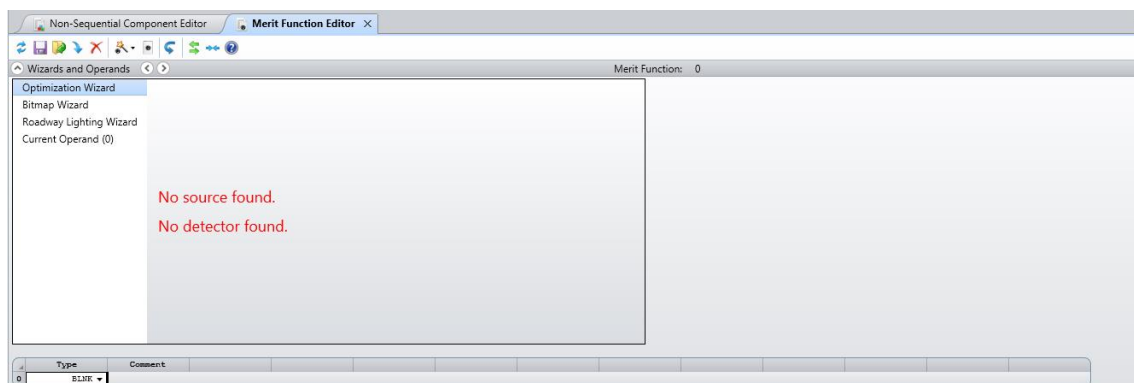
NSC Optimization Wizard

The Optimization Wizard is available in the Automatic Optimization section of the Optimize tab, or by expanding the bar labeled "Wizards and Operands" above the Merit Function Editor data. The Optimization Wizard displays a dialog box which allows selection of options for the auto merit function. This tool separates the options into detector settings, raytrace settings, and target computation settings.

The following screenshot was taken with the Zemax non-sequential scattering sample file, "Fluorescence Example.ZMX":
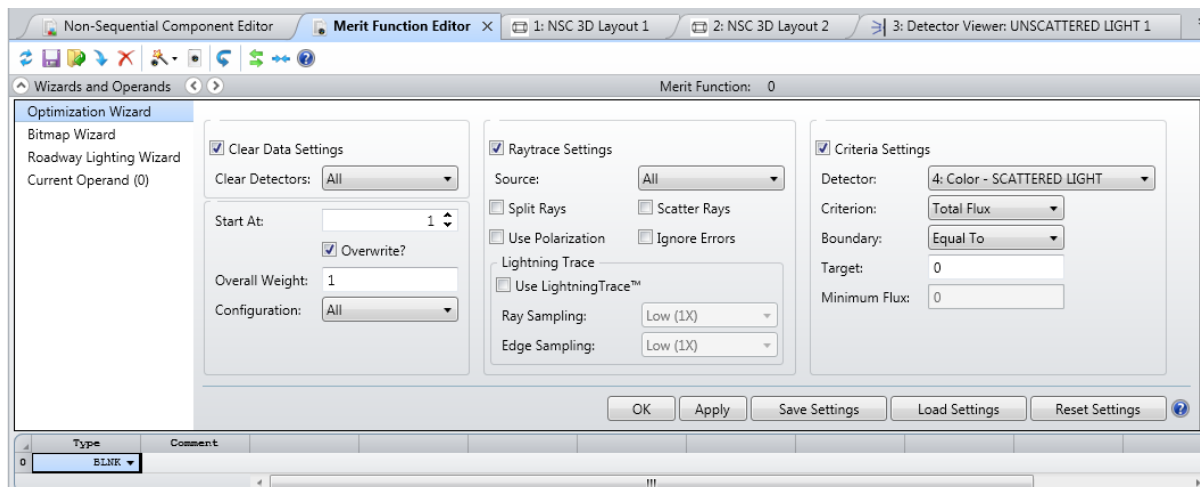
The merit function is a numerical representation of how closely an optical system meets a specified set of goals. Zemax uses a list of operands which individually represent different constraints or goals for the system such as total flux, uniformity, and many others.

The merit function is proportional to the square root of the weighted sum of the squares of the difference between the actual and target value of each operand in the list. The merit function is defined this way so a value of zero is ideal. The optimization algorithm will attempt to make the value of this function as small as possible, and so the merit function should be a representation of what you want the system to achieve.

The easiest way to define a merit function is to use the Non-Sequential Optimization Wizard. This shows the options for the auto merit function, which allows merit function to be built in a piecewise fashion using the **Apply** button. This will cause the dialog to stay open so that additional criteria may be added. The **OK** button will add the desired criteria and close the dialog.

Clear Data Settings

NSC merit functions must always contain one operand that clears all detectors before any data may be returned via other operands. The Optimization Wizard will add an operand to clear all detectors regardless of the clear detectors setting. This setting optionally clears individual detectors in the cases where complex tracing and clearing sequences are required in the merit function. If complex clearing isn't required, simply leave this setting at the default All.

Start At Settings

The auto merit function will begin at the start row specified. As operands are applied to the merit function, the start row will update to prevent overwriting previously created operands. If the overwrite checkbox is enabled, Zemax will delete all currently existing operands after the start row and define the current operands. If disabled, the additional lines will be

inserted at the start row and no lines will be overwritten. The overall weight defines the weight of operands added. The ability to define merit functions for all or individual configurations is available.

Raytrace Settings

All or individual sources can be raytraced depending on the source setting. Additionally, the ability to control splitting, scattering, polarization, and error ignoring are available. See "Ray Trace Control" of "The Analyze Tab" for more details on these settings. LightningTrace is available as an alternative to conventional raytracing. This feature is only available in Zemax-Premium. See "LightningTrace Control" of "The Analyze Tab".

Criteria Settings

Multiple different optimization criteria are supported. The available criteria depend on the detector type selected and include Spot Radius, Spot X, Spot Y, Angular Radius, Angular X, Angular Y, Spatial Uniformity, Angular Uniformity, Total Flux, Centroid X, Centroid Y, Angular X Centroid, and Angular Y Centroid. Only supported criteria types will be listed in the criterion dialog when selecting different detectors. For the widest range of available criteria use the detector rectangle. The criterion may be targeted to be equal to, greater than, or less than some specified target. The target indicates the desired value of the criteria in lens units.

In NSC optimization it is frequently necessary to maintain a minimum flux target to prevent Zemax from arriving at solutions with no rays landing on detectors. A minimum flux target can be specified in lens units and is generally recommended. This minimum flux target will be ignored if it is left at the default value of zero. As stated previously, the ability to create merit functions containing multiple types of targets is supported using the Apply button. Note that after Apply is clicked the first time, the clear data and raytrace settings are deactivated to prevent redundant definitions of these operands. In addition the Apply and OK buttons are deactivated until a change is made to the settings, also for the purpose of preventing redundant operand definition.

NSC Bitmap Wizard

The Bitmap Wizard is available in the Automatic Optimization section of the Optimize tab, or by expanding the bar labeled "Wizards and Operands" above the Merit Function Editor data.

The following screenshot was taken with the Zemax non-sequential scattering sample file, "Fluorescence Example.ZMX":
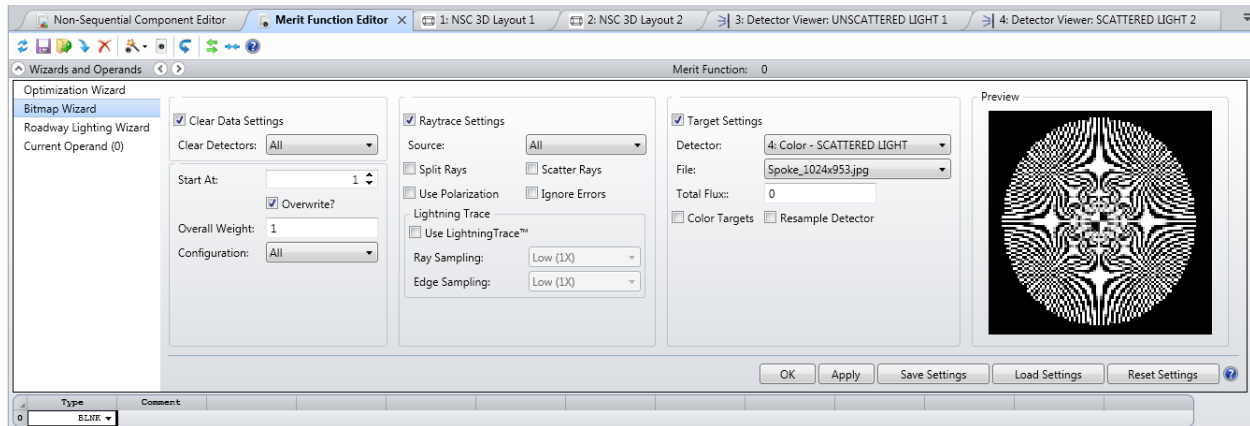
This feature is only available in the Premium edition of Zemax.

The NSC Bitmap Wizard allows the use of an image as the target for optimization. In many applications, the desired distribution of energy isn't something as simple as uniformity or maximum power, but a more complex pattern. Additionally, color targets may be desired in addition to flux targets. If the desired distribution of flux and color can be defined in a standard image file format, the bitmap merit function can convert this into a merit function. Currently, only spatial distribution targets are supported.

Clear Data Settings

These settings are identical to the NSC Optimization Wizard's clear data settings. See "NSC Optimization Wizard" under the "Automatic Optimization" section of "The Optimize Tab (Non-Sequential UI Mode)".

Start At Settings

These settings are identical to the NSC Optimization Wizard's Start At settings. See "NSC Optimization Wizard" under the "Automatic Optimization" section of "The Optimize Tab (Non-Sequential UI Mode)".

Raytrace Settings

All or individual sources can be raytraced depending on the source setting. Additionally, the ability to control splitting, scattering, polarization, and error ignoring are available. See "Ray Trace Control" of "The Analyze Tab" for more details on these settings. LightningTrace is available as an alternative to conventional raytracing. See "LightningTrace Control" of "The Analyze Tab".

Target Settings

This tool only supports color and rectangle detectors because the input image must be converted to spatial optimization targets. Consequently, these types of detectors are the only ones which will show up in the detector list. The file represents the target spatial distribution. Supported file formats are BMP, JPG, and PNG, and these files must be located in the <data>\IMAFiles folder. See "Folders" for more details.

While an image file indicates the desired relative flux distribution, it doesn't indicate the total flux. The user must specify this value in lens units using the total flux setting. If the detector selected is a detector color, the ability to define color targets is available. Otherwise, only flux will be targeted. Additionally, the input image can be scaled to the resolution of the detector or the detector resolution can be changed to match the image. The resample detector setting enables the latter. In all cases, a preview of the targeted image on the detector is shown. This allows the user to see exactly what Zemax will be targeting on the detector. Note that some aliasing effects may be present due to the size of the dialog preview, the resolution of the image, and the resolution of the screen. These artifacts are only present in the preview image, and not in the actual targets defined in the merit function.

High resolution images and detectors

The NSC Bitmap Wizard adds one operand per pixel if only flux is targeted and three operands per pixel if color and flux are targeted. For high resolution detectors or target images this can cause a very large number of operands to be required. For this reason, it is advisable to use the lowest resolution image that accurately represents the desired flux/color distribution. Once the user has clicked OK or Apply, Zemax will indicate the progress of merit function creation in the title bar of the dialog and enable the terminate button. If mistakes were made in the settings or the creation of operands is taking too long, the user can optionally terminate the process. Any operands already added will be deleted from the merit function.

NSC Roadway Lighting Wizard

The NSC Roadway Lighting Wizard is available in the Automatic Optimization section of the Optimize tab, or by expanding the bar labeled "Wizards and Operands" above the Merit Function Editor data.

The following screenshot was taken with the Zemax non-sequential scattering sample file, "Fluorescence Example.ZMX":
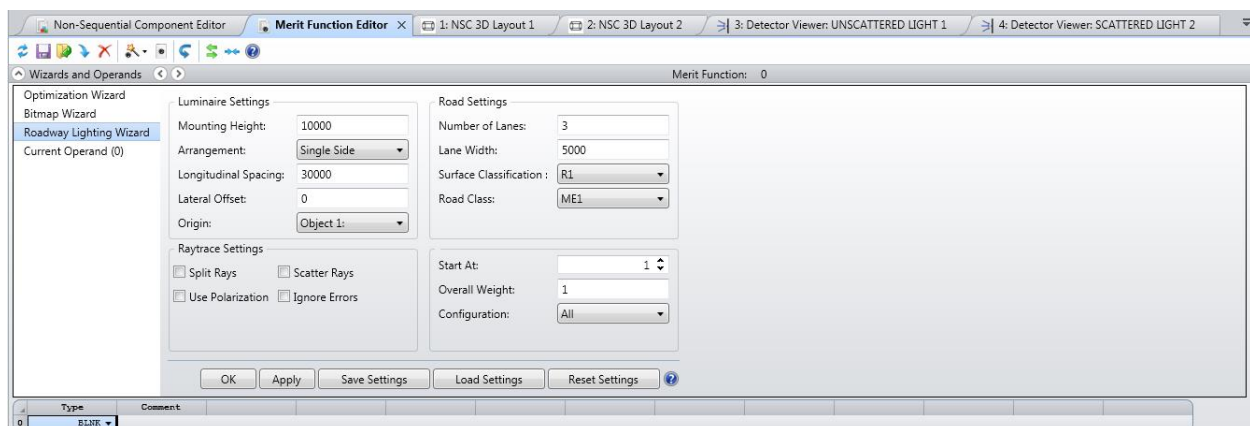
    This feature is only available in the Premium edition of Zemax.

The NSC Roadway Lighting Wizard is designed to optimize roadway lamps to pass the CIE 140-2000 specifications. For more details on these computations, see "Roadway Lighting" of "The Analysis Tab". Using this tool Zemax will add one NSTW operand to trace rays from all active sources and an NSRW operand for each of the eight specifications. Each

specification is a threshold, so OPGT and OPLT operands are used to apply boundary targets. See "NSTW" and "NSRW" for more details.

Luminaire Settings

These settings are identical to those used in the Roadway Lighting Analysis. See "Roadway Lighting" of "The Analysis Tab" for details.

Start At Settings

The merit function will begin at the start row specified and delete all existing operands after that line. The overall weight defines the weight of operands added. The ability to define merit functions for all or individual configurations is available.

Road Settings

These settings are identical to those used in the Roadway Lighting Analysis. See "Roadway Lighting" of "The Analysis Tab" for details.

Raytrace Settings

All or individual sources can be raytraced depending on the source setting. Additionally, the ability to control splitting, scattering, polarization, and error ignoring are available. See "Ray Trace Control" of "The Analyze Tab" for more details on these settings.

Remove All Variables (NSC)

See the description of "Remove All Variables" under the "Automatic Optimization" section of "The Optimize Tab".

Global Optimizers Group (NSC)

These are the global optimization tools available in Non-Sequential UI Mode.

Global Optimizer (NSC)

See the description of "Global Optimizer" under the "Global Optimizers" section of "The Optimize Tab".

Hammer Optimizer (NSC)

See the description of "Hammer Optimizer" under the "Global Optimizers" section of "The Optimize Tab".

The Tolerance Tab
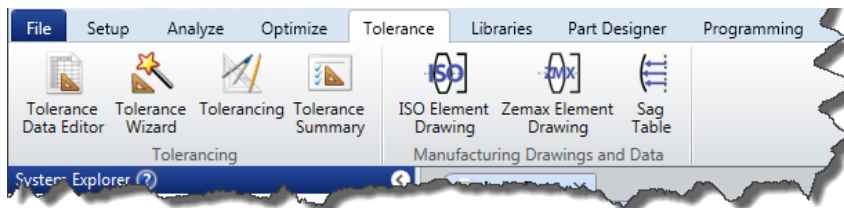
For more information, also see Tolerancing.

Tolerancing Group

Tolerance Data Editor

The Tolerance Data editor is used to define, modify, and review the system tolerance values.

See "Tolerancing" for details. Also see Tolerance Operands for a summary table and full list of tolerancing operands.

Tolerance Data Editor Tool Bar Options
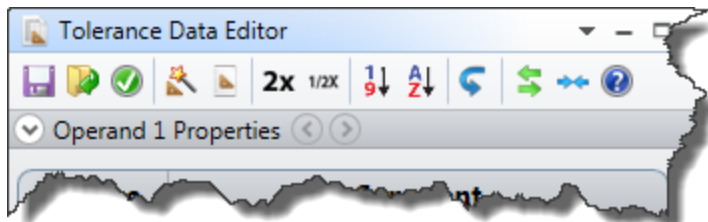
 Tolerance Save

Saves the current tolerance data in a *.TOL file. This step is only required if the data is to be subsequently loaded into another lens or stored for archival purposes. Zemax automatically stores the tolerance data with the lens when the entire lens is saved.

 Tolerance Load

Loads tolerance data previously stored in a *.TOL file or in a *.ZMX file. Either file type may be selected; only the tolerance data portion of the file will be loaded into the spreadsheet. The current tolerance data is destroyed.

 Validate

Checks all tolerance operands for proper surface ranges and other common errors.

 Tolerance Wizard

Invokes the default tolerances dialog box. See "Defining default tolerances"

 Tolerance Listing

Generates a test listing of the tolerances which can be saved or printed

 Loosen 2x

Increases all tolerance ranges by a factor of 2. This is a quick way of loosening the tolerances if they are all too tight.

1/2X **Tighten 2x**

Decreases all tolerance ranges by a factor of 2. This is a quick way of tightening the tolerances if they are all too loose.

Sort by Surface

Sorts all operands in ascending order first by surface number, then by type. The operands COMP and CPAR will always be placed at the top of the list. The SAVE operand will automatically be moved to stay just beneath the same operand it was beneath before sorting, because the SAVE operand is associated with the previous operand in the list. The STAT operand, if present, will be placed at the top of the list, and this operand must manually be moved or reinserted. Since STAT affects all operands which follow in the list, sorting the list will invalidate the STAT operand. Whenever STAT is used in the body of the tolerance list (to change the statistics on the fly) then editing of the list to correctly replace the STAT operand will be required whenever sorting. Note multiple STAT operands may be required if the operands that originally followed the STAT operand are dispersed through the list by the sort operation.

Sort by Type

Sorts all operands in ascending order by type, then by surface number.

Go To Operand

Go to the specified operand number.

Reset Column Order

Resets column order to default settings

Reset Column Widths

Resets column widths to standard size

Help

See more help on this feature

Right Click Menu Options
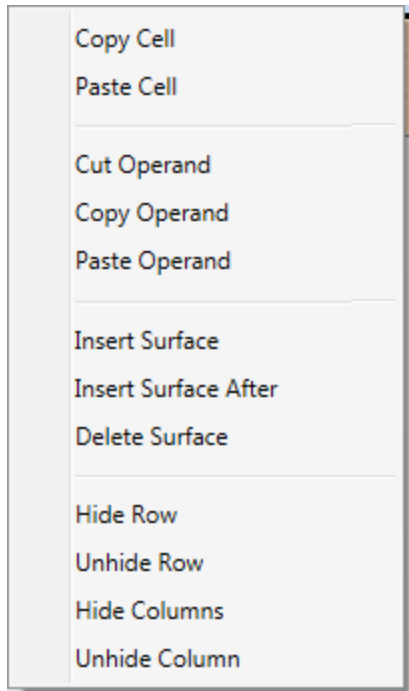
Copy Cell: Copies a single cell's data to the Windows Clipboard.

Paste Cell: Pastes the data for a single cell from the Windows Clipboard to the current cell. The data must first have been copied to the Windows Clipboard using "Copy Cell" described above.

Cut Operands: Copies all the data for a single operand or a range of operands to the Windows Clipboard, then deletes the operands. The operand or range of operands must first be selected using either of the following techniques:

Using the mouse: Click on the first operand to be selected. Hold down the left mouse button, and drag the mouse cursor to cover the range of operands desired. The selected operands will be highlighted in inverse colors. To select only one operand, drag the mouse up or down from the operand desired until two operands are selected, then move the mouse back to the desired operand.

Using the keyboard: Move the cursor to any cell on the desired operand row. Then hold down the shift key, while moving the cursor up or down until the desired range of operands is selected. The selected operands will be highlighted in inverse colors. To select only one operand, move the cursor up or down from the operand desired until two operands are selected, then move the cursor back to the desired operand.

Copy Operands: Copies all the data for a single operand or a range of operands to the Windows Clipboard. To select a single operand or a range of operands, see the discussion under "Cut Operands" above.

Paste Operands: Copies all the data for a single operand or a range of operands from the Windows Clipboard to the current cursor location in the Tolerance Data Editor. The operand data must first have been copied to the Windows Clipboard using either "Cut Operands" or "Copy Operands" described above.

Insert Operand: Inserts a new row in the spreadsheet at the current row. The Insert key will perform this same function.

Insert After: Inserts a new row in the spreadsheet after the current row. The Ctrl-Insert key combination will perform this same function.

Delete Operand: Deletes the row at the current cursor location. The Delete key will perform this same function.

Hide Row: Minimizes the width of the row displaying the cursor.

Unhide Row: Sets hidden row to the default width

Hide Column: Minimizes the width of the column displaying the cursor.

Unhide Column: Sets hidden columns to the default width

Operand Properties
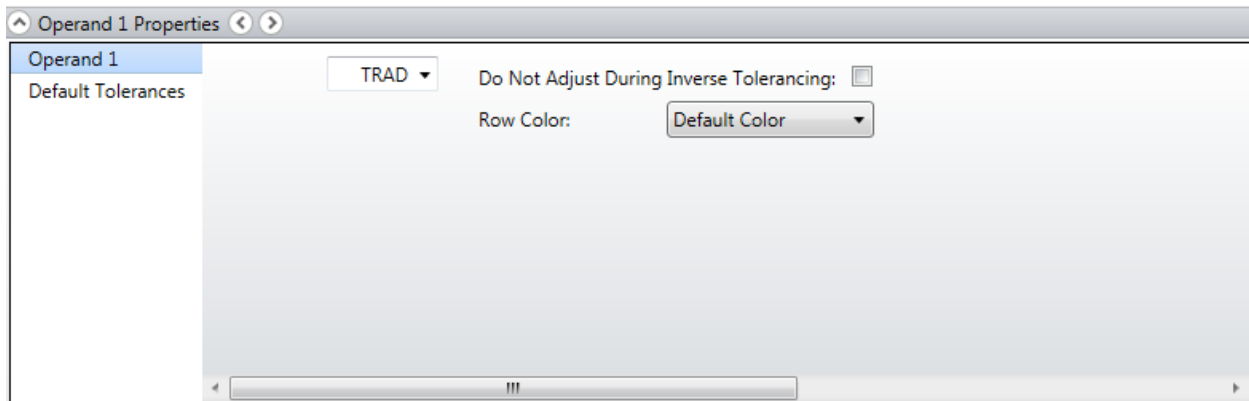
Operand Type: This option allows the type of operand and other data to be changed. For a complete description of tolerance operands, see Tolerance operands.

Do Not Adjust During Inverse Tolerancing: Inverse tolerancing can be disabled for individual operands; which prevents the inverse tolerance algorithm from tightening tolerances even if the estimated performance exceeds the specified limits.

Row Color: Selects either default or none for the color for each row in the spreadsheet. Individual row colors may be set using the row color setting on the operand type dialog box.

Summary of Tolerance Operands

TRAD   Tolerance on surface radius of curvature in lens units

TCUR   Tolerance on surface curvature in inverse lens units

TFRN   Tolerance on surface radius of curvature in fringes

TTHI   Tolerance on thickness or position in lens units

TCON   Tolerance on conic constant (dimensionless)

TSDI   Tolerance on semi-diameter in lens units

TSDX   Tolerance on Standard surface x-decenter in lens units

TSDY   Tolerance on Standard surface y-decenter in lens units

TSTX   Tolerance on Standard surface tilt in x in degrees

TSTY   Tolerance on Standard surface tilt in y in degrees

TIRX   Tolerance on Standard surface tilt in x in lens units

TIRY   Tolerance on Standard surface tilt in y in lens units

TIRR   Tolerance on Standard surface irregularity

TEXI   Tolerance on surface irregularity using Zernike Fringe polynomials

TEZI   Tolerance on surface irregularity using Zernike Standard polynomials

TPAI   Tolerance on the inverse of parameter value of surface

TPAR   Tolerance on parameter value of surface

TEDV   Tolerance on extra data value of surface

TIND   Tolerance on index of refraction of surface

TABB   Tolerance on Abbe number of surface

TCMU   Tolerance on coating multiplier of surface

TCIO   Tolerance on coating index offset of surface

TCEO   Tolerance on coating extinction offset of surface

TEDX   Tolerance on element x-decenter in lens units

TEDY   Tolerance on element y-decenter in lens units

TETX   Tolerance on element tilt about x in degrees

TETY   Tolerance on element tilt about y in degrees

TETZ   Tolerance on element tilt about z in degrees

TUDX   Tolerance on user defined x-decenter in lens units

TUDY   Tolerance on user defined y-decenter in lens units

TUTX   Tolerance on user defined tilt about x in degrees

TUTY   Tolerance on user defined tilt about y in degrees

TUTZ   Tolerance on user defined tilt about z in degrees

TNPS   Tolerance on NSC object position.

TNPA   Tolerance on an NSC object parameter data

TMCO   Tolerance on multi-configuration editor value

CEDV   Sets an extra data value as a compensator

CMCO   Sets a multi-configuration value as a compensator

COMM   This operand is used to print a comment in the tolerance analysis output report

COMP   Sets a thickness, radius or conic compensator.

CPAR   Sets a parameter as a compensator

SAVE   Saves the file used to evaluate the tolerance on the prior row in the editor

SEED   Seeds the random number generator for MC analysis. Choose 0 for random.

STAT   Defines statstics used 'on the fly' during Monte Carlo analysis

TWAV   Sets the test wavelength for operands measured in fringes

For a complete description of tolerance operands, see Tolerance operands.

Tolerance Wizard

The default tolerances may be defined by selecting the Tolerance Wizard.
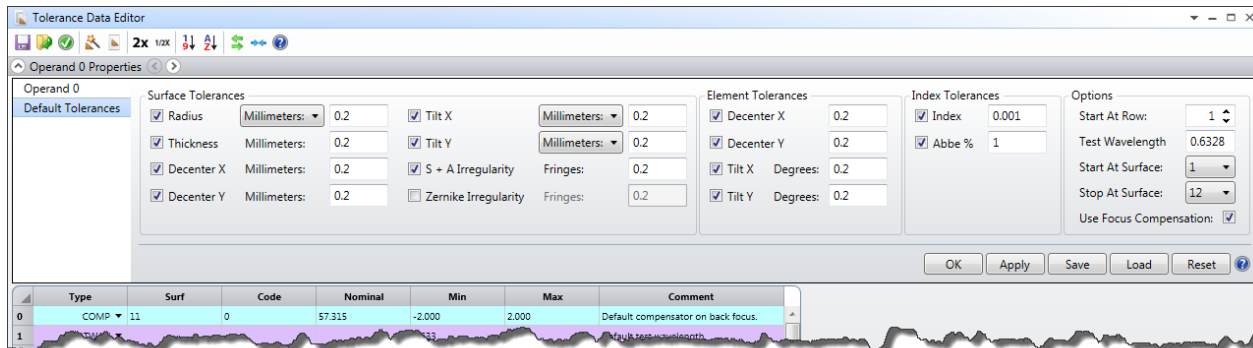
The default tolerances dialog box consists of several sections grouped by tolerance type.

Surface tolerances

Radius: If this box is checked, then the default radius tolerances will be included. The default tolerances may be specified by a fixed distance in lens units or by fringes of power at the test wavelength (defined by the TWAV operand). This tolerance is only placed on surfaces which have optical power, which excludes dummy surfaces which have the same index on both sides. If the surface is plano, then the default tolerance is specified as a variation in fringes, even if another option is selected.

Thickness: If checked, a thickness tolerance is specified on each vertex separation. It is assumed that all variations in thickness affect only that surface and any surfaces in contact with that element; therefore, the first air space after the thickness is used as an adjuster. See the detailed discussion in "TTHI: Tolerance on thickness" .

Decenter X/Y: If checked, decenter tolerances are added to each individual lens surface. Tolerances are defined as a fixed decenter amount in lens units. Zemax uses TSDX and TSDY for Standard surface decenters, and TEDX and TEDY for non-Standard surfaces.

Tilt (TIR) X/Y: If checked, a tilt or "total indicator runout" tolerance in either lens units or degrees is added to each lens surface. Zemax uses TSTX and TSTY for Standard surface tilts in degrees, TIRX and TIRY for Standard surface total indicator runout (tilt or wedge) in lens units, and TETX and TETY for non-Standard surface tilts in degrees. If lens units are selected, Zemax automatically converts the dimension to degrees of tilt if required for the TSTX/TSTY or TETX/TETY operands. The conversion from total indicator runout in lens units to tilt is given by

$$\theta_x = \frac{\Delta_y}{2S}, \text{ where}$$

S is the semi-diameter of the surface and Δ is the total indicator runout, with a similar expression for θ_y . This expression assumes the angles are relatively small. Note that a length of total indicator runout in the x direction corresponds to a tilt about the y axis, and a length of total indicator runout in the y direction corresponds to a tilt about the x axis.

S + A Irreg: If checked, a spherical and astigmatism irregularity is specified on each Standard surface type. For details, see "TIRR: Tolerance on surface irregularity" .

Zernike Irregularity: If checked, a Zernike irregularity is specified on each Standard surface type. For details, see "TEZI: Tolerance on surface irregularity using the Standard Zernike model" . By default, Zemax uses Zernike terms 2 through 45.

Element tolerances

Decenter X/Y: If checked, decenter tolerances are added to each lens group. Tolerances may be defined as a fixed decenter amount in lens units.

Tilt X/Y: If checked, a tilt tolerance in degrees is added to each lens group and surface. It is important to note that lens groups are by default tipped about the vertex of the first surface in the group. See the section "TETX, TETY, TETZ: Tolerance on element tilts" for information on tipping about some other point.

Index tolerances

Index: TIND is used to model changes in the index of refraction. The units are change in relative index.

Abbe: TABB is used to model changes in the Abbe number. The units are percentage of the catalog Abbe value.

Other controls

Start At Row: This control indicates where in the Tolerance Data Editor the default tolerances should be placed. If the row number is greater than 1, then the new default tolerances will be appended starting at the specified row number. If the row number is -1, then the tolerances will be appended to the end of the current list.

Test Wavelength: The wavelength, in micrometers, used to measured fringes of power or irregularity.

Use Focus Comp: If checked, then a default compensator of the back focus (the thickness prior to the image surface) will be defined. Using at least one compensator can greatly relax certain tolerances, however whether or not compensators should be used depends upon the specifics of the design. Other compensators may be defined. See the Section "Defining compensators" for more information.

Start / Stop At Surface: These controls define the range of surfaces to apply the default tolerances to. There are also six buttons:

OK: Accept these settings and generate the default tolerances. Cancel: Close the dialog box without changing the default tolerances. Save: Save these settings for future use.

Load: Restore the previously saved settings. Reset: Restore the settings to the default values. Help: Invoke the help system.

By default, the Monte Carlo analysis that Zemax performs draws random values from a Gaussian "normal" distribution. Once the default tolerances are defined, they are saved with the lens file automatically. If additional surfaces are inserted in the Lens Data Editor, the tolerance surfaces are renumbered automatically.

Tolerancing
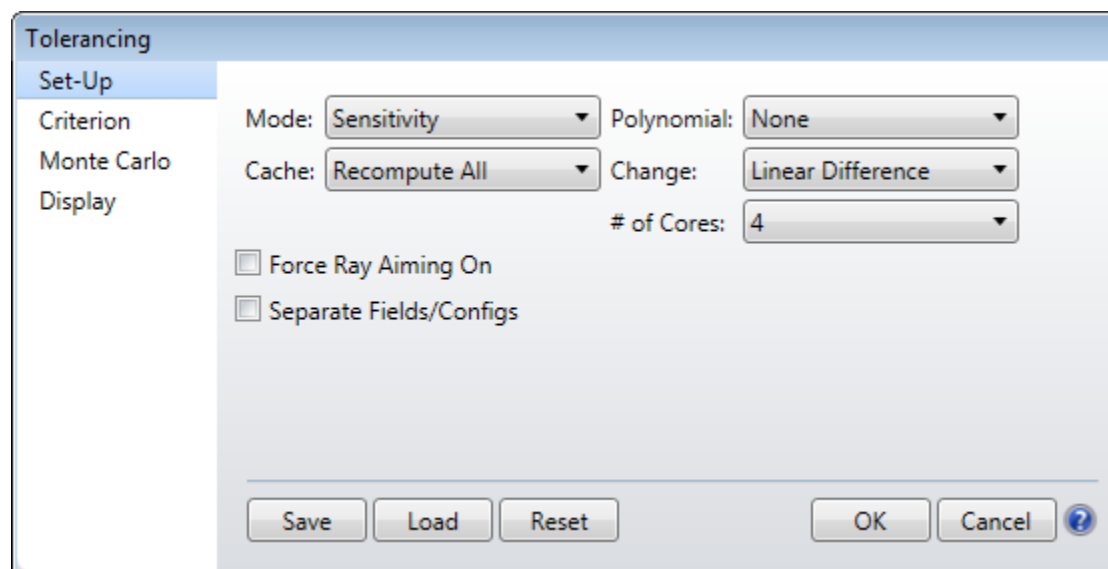
Once all the tolerance operands and compensators have been defined, the tolerance analysis may be performed. To perform a tolerance analysis, select Tolerancing from the Tools menu in the main program menu bar.

The dialog box which appears has several controls which are described below.

<u>Set-Up tab</u>

Mode: Four modes are supported: Sensitivity, Inverse Limit, Inverse Increment, and Skip Sensitivity.

- Sensitivity mode computes the change in the criterion for each of the extreme values of the tolerances. Inverse Limit computes the value of each tolerance that will yield a criterion being equal to the value specified by the Limit parameter. The Limit parameter is only available if the mode is

- Inverse Limit. Inverse mode will change the min and max values of the tolerance operands. See "Inverse sensitivity analysis" .

- Inverse Increment computes the value of each tolerance that will yield a change in the criterion equal to the value specified by the Increment parameter. The Increment parameter is only available if the mode is Inverse Increment. Inverse mode will change the min and max values of the tolerance operands. For a description of how change is defined, see "Change" below. See "Inverse sensitivity analysis" .

- Skip Sensitivity will bypass the sensitivity analysis and proceed to the Monte Carlo analysis.

Polynomial: The Polynomial feature has three options: None, 3-Term, and 5-Term. If either 3-Term or 5-Term is selected, the tolerance sensitivity will be determined at 4 and 6 points, respectively, spanning the tolerance

perturbation range and a polynomial fit of the resulting criterion as a function of the tolerance perturbation will be computed and displayed. The 5-Term polynomial is:

$$P = A + B\delta + C\delta^2 + D\delta^3 + E\delta^4,$$

where δ is the tolerance perturbation value and P is the resulting criterion. The 3-Term polynomial omits the D and E terms. Because the polynomial fit requires evaluation of the criterion and compensators at multiple points, selecting this option increases the computation time. However, the polynomial values are cached, and once computed, can be used to greatly speed up subsequent tolerance runs with modified tolerance values.

See Cache below for details.

Cache: The cache feature can be used to greatly speed up the sensitivity and inverse sensitivity tolerance analysis, but must be used with some care. The first time the tolerance analysis is run, all the tolerances, perturbed criterion, and polynomial fit data (if any) is saved in memory. This cached data may be used to quickly generate the tolerance analysis again. If "Recompute All" is selected, the cached values are ignored, and all the tolerances are computed in full, and the new values stored in the cache. If "Recompute Changed" is selected, then only those tolerances that have been modified are recomputed. Results for unmodified tolerances are extracted from the cache rather than computed, significantly speeding up the analysis. If "Use Polynomial" is selected, than the last polynomial fit computed for each tolerance is used to quickly estimate the perturbed values. Zemax will automatically invalidate the cached values if any of the tolerance settings, such as criterion or Sampling, are modified. However, if any significant changes are made to the lens file itself, such as modifying surface data, the cached tolerances may be incorrect but Zemax is unable to detect this. Therefore, the cache feature should only be used in a single session where no changes are being made to the lens. The intention of this feature is to significantly speed up a sequence of tolerance analysis where changes are being made to the values of individual tolerances, but no other changes are being made between tolerance runs. If there is ever a doubt as to the validity of the cached data, set "Cache" to "Recompute All" and repeat the tolerance analysis. When using the cache, compensator data is not displayed (although the data presented does consider the affects of compensation), and separation by fields and configurations is not supported.

Change: Change determines how the change in the criterion and the predicted performance are computed. If

Linear Difference is selected, then the change due to a tolerance is computed as

$\Delta = P - N$,

where P is the perturbed criterion and N is the nominal criterion. If Change is set to RSS Difference, the change is computed as:

$$\Delta = S(P - N) \sqrt{|P^2 - N^2|} \, ,$$

where the function S(x) returns +1 if the x ≥ 0 and -1 otherwise.

Force Ray Aiming On: If the lens being toleranced is already using ray aiming, then ray aiming will be used when evaluating the tolerances. If ray aiming is not already on, then ray aiming will only be used if this box is checked. Generally, using ray aiming yields more accurate results but slower computation speed. For preliminary or rough tolerance work, leave the switch at the default "off", but for final or precise work, set the switch "on".

 # CPU's: Selects the number of CPU's over which to spread the tolerance analysis task. More than 1 may be selected, even on a single CPU computer, in which case the single CPU will time share the multiple simultaneous tasks. The default is the number of processors detected by the operating system.

Separate Fields/Configs: If checked, the criterion will be computed and displayed for all field positions in all configurations individually. If unchecked, the criterion will be computed as an average over all field positions in all configurations. During inverse tolerancing, if Separate Fields/Configs is unchecked, the inverse analysis is done on the overall tolerance criterion, which is usually an average of performance over all fields in all configurations. The problem with using the average performance is that some fields or configurations may be significantly degraded by tolerance defects while other fields or configurations are not, and the average may not reveal the severity of the loss of performance at a few fields or configurations. If this option is checked, then Zemax computes the criterion at each field in each configuration individually, and verifies that each field meets either the Limit or Increment value. For Inverse Increment mode, Zemax computes the nominal performance at every field position and reduces the tolerances until the criterion at every field is degraded no more than the increment value. See "Inverse sensitivity analysis" .

Criterion tab

Criterion: This control is used to specify what shall be used as the criterion for tolerancing. The options are:

- RMS spot size (radius, x, or y): The best choice for systems which are not close to the diffraction limit; for example, systems with more than one wave of aberration. This is the fastest option. Zemax always uses a centroid reference for tolerancing.

- RMS wavefront: The best choice for systems which are close to the diffraction limit; for example, systems with less than one wave of aberration. This is nearly as fast as RMS spot radius. Zemax always uses a centroid reference for tolerancing.

- Merit Function: Uses whatever merit function has been defined for the lens. This is useful for user-defined tolerancing criterion. User-defined merit functions may also be required for systems with non-symmetric fields, or with significant surface apertures which remove rays. If the user-defined merit function is used, no boundary constraints on the compensators are automatically added to the merit function. If the merit function was generated by Zemax as one of the default merit functions, make sure the "Assume Axial Symmetry" option was checked OFF, see "Assume Axial Symmetry" for a discussion. Separate Fields/Configs is not supported when using this criterion.

- Geometric or Diffraction MTF (average, tangential, or sagittal): The best choice for systems which require an MTF specification. If average is selected, the average of the tangential and sagittal responses is used. Diffraction based MTF tolerancing can be problematic if the tolerances are loose, because the diffraction MTF may not be computable or meaningful if the OPD errors are too large. This is especially true if the spatial frequency is high enough and the performance poor enough for the MTF to go to zero at some frequency below

the frequency being analyzed. MTF is the slowest of the default criterion. The frequency at which the MTF is computed is specified in the "MTF Frequency" control.

- Boresight error: Boresight error is defined as the radial chief ray coordinate traced for the on axis field divided by the effective focal length. This definition yields a measure of the angular deviation of the image. Zemax models boresight error by using just one BSER operand (see the Optimization chapter for details on BSER). Any element or surface decenters or tilts will tend to deviate the chief ray and increase the values of the BSER operand. Boresight error is always computed at the primary wavelength in radians. Boresight error should only be used with radially symmetric systems. Note boresight error gives no indication of image quality; it is a measure of the deviation of the axis ray.

- RMS Angular radial, x, or y aberrations. This tolerance criterion is best used for afocal systems (see "Afocal Image Space"). The angular aberrations are based upon the direction cosines of the output rays.

User Script: A user script is a macro-like command file which defines the procedure to be used for alignment and evaluation of the lens during tolerancing. For details on this option, see the section "Using Tolerance Scripts" . Separate Fields/Configs is not supported when using this criterion.

Limit: When using Inverse Limit mode, this control is active and is used to define the limit on the criterion for computing inverse tolerances. For example, suppose the Change is Linear Difference, Criterion is RMS Spot Radius, and the nominal RMS of a system is 0.035. If Limit is set to 0.050, then Zemax will compute the min and max value of each tolerance that degrades the performance to an RMS of 0.050. The Limit value must represent worse performance than the nominal system has. When using MTF as a merit, then Limit is the lower bound on MTF since lower numbers indicate worse performance. See "Inverse sensitivity analysis" . The nominal value for the currently selected criterion may be computed by pressing the "Check" button adjacent to the Limit edit window.

Increment: When using Inverse Increment mode, this control is active and is used to define the limit on the change in the criterion for computing inverse tolerances. For example, suppose the Change is Linear Difference, Criterion is RMS Spot Radius, and the nominal RMS of a system is 0.035. If Increment is set to

0.01, then Zemax will compute the min and max value of each tolerance that degrades the performance to anRMS of 0.045. The Increment value must be positive to represent a degradation in performance. When using MTF as a merit, the Increment is still positive, and Zemax automatically interprets this number as a decrease in MTF from the nominal. See "Inverse sensitivity analysis" . The nominal value for the currently selected criterion may be computed by pressing the "Check" button adjacent to the Increment edit window.

Sampling: Sampling is used to set how many rays are traced when computing the tolerance criterion. Higher sampling traces more rays, and gives more accurate results. However, the execution time increases. If the selected criterion is RMS spot or RMS wavefront, then the sampling value is an integer that refers to the number of rays traced along a radial arm of the pupil in the Gaussian quadrature technique (see "Selecting the pupil integration method" for a description of this technique). The number of arms is always twice the number of rays along each arm. If MTF is the selected criterion, then the sampling refers to the pupil grid size, with a sampling of 1 yielding a 32 x 32 grid, sampling of 2 yielding a 64 x 64 grid, etc. Usually, a sampling of 3 or 4 is sufficient for quality optical systems. Systems with high amounts of aberration require higher sampling than systems with low aberration. The most reliable method for determining the best sampling setting is to run the tolerance at a sampling of 3, then again for a sampling of 4. If the results change moderately, then use the higher setting. If they change substantially, check the next higher sampling setting. If the results change little, go back to the lower sampling. Setting the sampling higher than required increases computation time without increasing accuracy of the results.

MTF Frequency: If MTF is selected as the Merit, then this control is active and is used for defining the MTF frequency. MTF frequency is measured in MTF Units, see "MTF Units".

Comp: This control determines how the compensators are evaluated. "Optimize All" will use the optimization capability of Zemax to determine the optimum values of all defined compensators. Although optimization is accurate, it is slow to execute. There are two different optimization algorithms available, DLS and OD. If "Optimize All (DLS)" is selected, Zemax will execute one cycle of the Orthogonal Decent algorithm then execute the Damped Least Squares algorithm. If "Optimize All (OD)" is selected, Then the Orthogonal Descent algorithm only is used. For more information, see "Performing an optimization" . If "Paraxial Focus" is selected, only the change in paraxial back focus error is considered as a compensator; all other compensa- tors are ignored. Using Paraxial Focus is very useful for rough tolerancing, and is

significantly faster than using "Optimize All". If "None" is selected, no compensation will be performed, and any defined compensators will be ignored.

Config: For multi-configuration lenses, indicates which configuration should be used for tolerancing. The selected configuration only will be considered, and the configuration number will be printed on the final report. If "All" is selected, then all configurations will be considered at once.

Fields: Generally speaking, the field definitions used for optimization and analysis are inadequate for tolerancing. for example, a rotationally symmetric lens may use field definitions of 0, 7, and 10 degrees. For tolerancing purposes, the lack of symmetry in the field definitions may cause inaccurate results when analyzing tilt or decenter tolerances. When constructing a merit function to use for tolerancing Zemax can use three different field settings:

- Y-Symmetric: Zemax computes the maximum field coordinate, then defines new field points at +1.0, +0.7, 0.0, -0.7, and -1.0 times the maximum field coordinate, in the Y direction only. All X field values are set to zero. This is the default for rotationally symmetric lenses.

- XY-Symmetric: Similar to Y-Symmetric, except there are 9 field points used. The 5 Y-Symmetric points are used, and -1.0, -0.7, +0.7, and +1.0 are added in the X axis direction only.

- User Defined: Use whatever field definitions exist in the current lens file. This option is required when using vignetting factors, tolerancing multiple configuration lenses, or using tolerance scripts. It is also highly recommended when tolerancing non rotationally symmetric lenses or lenses with complex field weighting that user defined fields be used.

If user defined fields are used, no adjustment of the weights is performed. For the Y-Symmetric case, the center point has a weight of 2.0, all others have a weight of 1.0. For the XY-Symmetric case, the center point has a weight of 4.0, all others unity.

Cycles: This determines how rigorously Zemax will attempt to optimize compensator values. If set to Auto, then Zemax will call the optimizer in "Auto" mode, which will run the optimizer until the optimization of the compensators has converged. For rough tolerancing, a low number, such as 1, 2, or 3, may be used. If the compensators are difficult to optimize, a higher setting may increase accuracy. If too few optimization cycles are selected, then the tolerances will be pessimistic; the predicted performance will be worse than the actual performance. The "Auto" setting is the safest to use. Higher settings increase accuracy at the expense of run time. This setting is only used if Comp is set to "Optimize All".

Script: The name of the script, if using the User Script criterion. User scripts must be text files ending in the extension TSC and be in the <data>\Tolerance folder (see "Folders").

Status: This control is used by the tolerance algorithm to provide status messages during computation of the nominal criterion when "Check" is pressed.

Monte Carlo tab

 # Monte Carlo Runs: This control is used to specify how many Monte Carlo simulations should be performed. The default setting of 20 will generate 20 random lenses which meet the tolerances specified. See the section "Monte Carlo simulations" for more details. The number of Monte Carlo runs may be set to zero, which will omit the Monte Carlo analysis from the summary report.

 # Monte Carlo Save: This option is used to save a specific number of lens files generated during the Monte Carlo analysis. The value specifies the maximum number of lens files to be saved. For example, suppose 20 is selected. After the first Monte Carlo lens is generated, the lens file will be saved in the file "MC_T0001.ZMX". The second Monte Carlo lens file will be generated, then saved in "MC_T0002.ZMX", and so on. Only the first

20 Monte Carlo lenses will be saved (the last will be "MC_T0020.ZMX"). If fewer than 20 Monte Carlo runs were requested, then fewer than 20 lenses would be saved. Be sure you have no lens files with the names "MC_Txxxx.ZMX", as Zemax will overwrite these files without warning as the lenses are saved. The purpose of this feature is to allow further study on the lenses being generated by the Monte Carlo feature. Note that the file names given here assume that no File Prefix to the Monte Carlo file names was defined; see File Prefix for details.

Statistics: Choose either a Gaussian "normal" distribution, "uniform", or "parabolic" distribution. This setting is only used by the Monte Carlo analysis; see that discussion for details on the statistics mode as well as the "STAT" command which provides detailed control over the statistics model used.

File Prefix: If a prefix string is provided, then the string will be prepended to the names of the Monte Carlo files generated. For example, if the prefix string is "Fast_Doublet_" then the first Monte Carlo file saved will be file name "Fast_Doublet_MC_T0001.ZMX", and similar names for the remaining saved files.

Save Best and Worst Monte Carlo Files: If checked, the best and worst Monte Carlo lens files generated will be saved. The file names are "prefixMC_BEST" and "prefixMC_WORST" where "prefix" is the File Prefix string.

Overlay MC Graphics: If checked, each open analysis graphical window (such as a ray fan or MTF graph) will be updated and overlayed for each Monte Carlo generated lens. The resulting plots are useful for showing the total range of performance for the simulated lenses. Analysis graphs which do not automatically change scale and do not depend upon surface numbers are the most useful, such as MTF, MTF vs. Height, Encircled Energy, and other plots that allow a user defined fixed scale, such as ray fan plots. Plots which change scale dynamically or require a fixed range of surfaces, such as layout plots, will not work reliably and are not supported. Static, Text, and Editor windows are not updated. Overlayed graphic windows will be flagged as static after the tolerance analysis is complete. The time to compute each of the analysis graphics for each MC lens obviously will slow down the tolerance analysis.

Display tab

Show Descriptions: If checked, a full description of the meaning of each tolerance operator will be provided in the analysis report. If unchecked, only the tolerance operator abbreviations will be listed.

Show Compensators: By default, the compensator values are not printed out during the sensitivity analysis. If this box is checked, then each compensator value will be printed along with the change in the criterion for each tolerance.

Hide All But Worst: If checked, printing of all the sensitivity data will be turned off. This is useful for decreasing the size of the output report. The "hide" check box is normally used in conjunction with the "Show worst" control.

Show worst: The "Show worst" control can be set to sort and display only some number of tolerance operands; this permits limiting the printout to the most severe tolerances only.

Output To File: If a valid file name is provided, the tolerance analysis output window text will be saved to the specified file. The path is always the same as the current lens file. Only provide a file name without a path in this box, such as OUTPUT.TXT. Leave this control blank to not save the output text.

Other buttons

There are also six buttons on the bottom of the dialog box:

Save: Save the currently selected tolerance options in a user selected file name for future use. The file ends in the extension TOP and these options may be used by other features.

Load: Restores tolerance options from a previously saved file.

Reset: Restore the settings to the default values.

OK: Performs the tolerance analysis using the current options.

Cancel: Exits the dialog box without performing the tolerance analysis, restoring the previous settings.

Apply: Saves the changed options for the session, even if Cancel is subsequently selected.

Once all of the options have been selected, press OK to begin the tolerance analysis. Details on the method of calculation are provided in the next section.

Tolerance Summary

This function generates a text listing of the tolerances which can be saved or printed. The format is somewhat easier to read than the Tolerance Listing, and it uses no mnemonics which makes the tolerances easier to understand for fabricators and others not familiar with Zemax terminology.
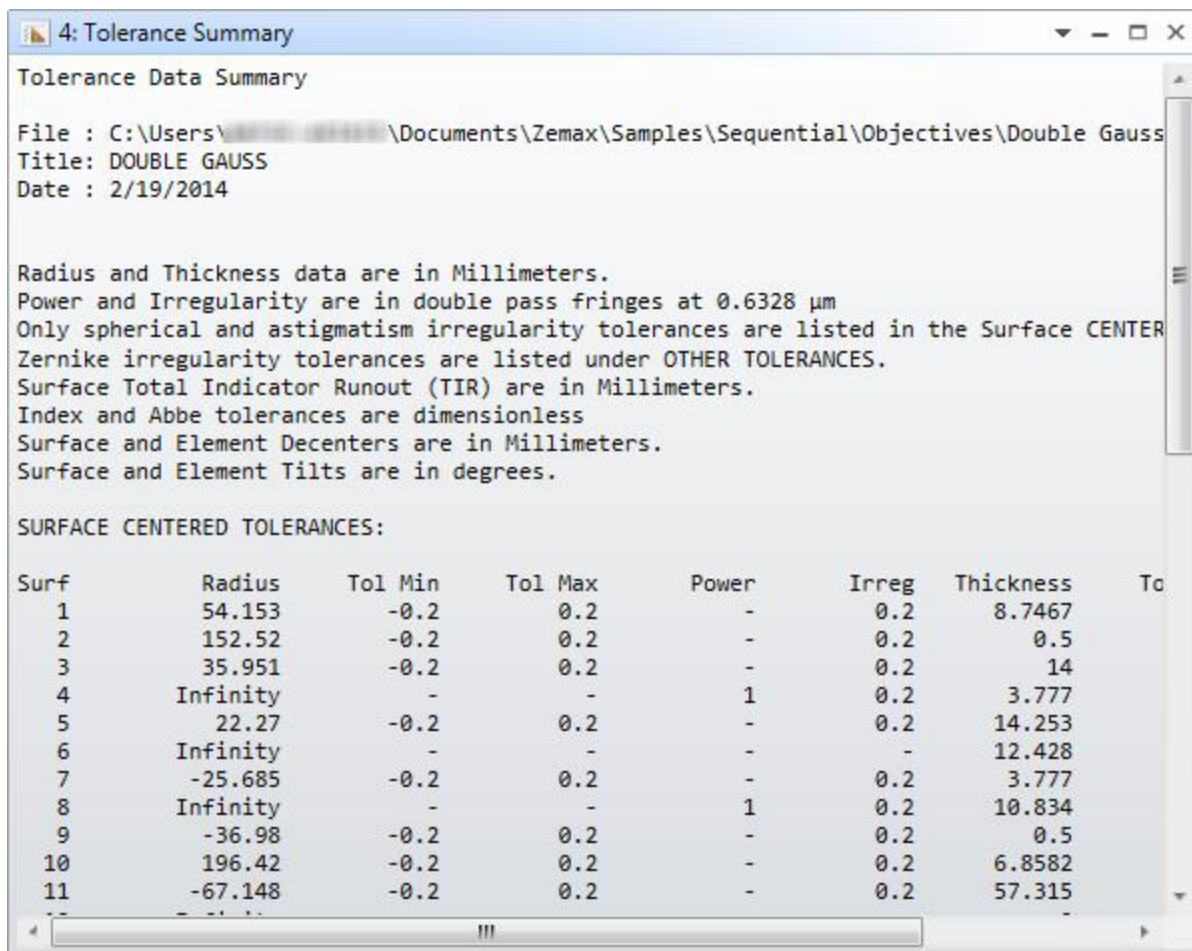
Manufacturing Drawings and Data Group

Zemax Element Drawing

This feature creates a mechanical drawing of surface, singlet, cemented doublet, or cemented triplet elements suitable for use in optical shop fabrication.

Surface The first surface of the element to be drawn.

Show As Select either "Surface", "Singlet", "Doublet", or "Triplet".

Note File Name The name of the ASCII file which contains the notes to be appended to the notes section of the element drawing. Notes should always start at number 2, since number 1 is reserved for the units specification.

Edit Note File Clicking on this button will invoke the Windows NOTEPAD.EXE editor, which can then be used to modify the selected note file.

Radius The radius tolerance value for the specified surface.

Irregularity The irregularity tolerance value for the specified surface.

Clear Aper The clear aperture diameter of the specified surface.

Thickness The center thickness tolerance of the specified surface.

Scale Factor If the scale factor is set to zero, then "Fill Frame" will be selected, which will scale the element to fill the right half of the element drawing. If a numeric value is entered, then the plot will be drawn in "real" scale, times the scale factor. For example, a scale factorof 1.0 will plot the element at actual size on the printer (not the display). A factor of 0.5 will plot the element at half scale.

Decimals The number of decimals to use in numeric values. Press the "Reset all but surface, show as, and titles" button to reformat all displayed numbers.

Drawing Title This field is for any user defined text. The default is the lens title.

Date If left blank, the default date format is used. If any text is supplied, the exact text is used.

Drawing Name All of these fields are for user defined text. Any text may be entered. No default value.

Approved

Revision

Drawn By

Project

Note Font Size Choose Standard, Medium, Small, or Fine. These are in order of decreasing font size. The Note Font Size setting only affects the size of the note file that is annotated on the drawing. Smaller fonts permit larger note files to be displayed.

Save As/Load From These two buttons allow saving and reloading of all settings displayed on the element drawing settings box. The data is stored in a binary format with the extension ELE.

Reset all but... If selected, this button will reset all the default tolerances and apertures for the specified surfaces, but the current surface, show as, and text titles will remain as they are.

Discussion The element drawing settings may be stored for the specific lens file by pressing the Save button. Unlike most analysis features, the element drawing feature saves all the settings for each surface separately. For example, the notes and tolerances for surface 1 may be saved, and then new notes and tolerances for surface 3 may be entered and then saved. To recall the settings for any specific surface, change the surface number to the desired surface, and then press the Load button. If a match is found with a previously saved surface, the settings for that surface will be displayed. This feature makes it easy to regenerate complex drawings for multiple element systems.

An important feature of the element drawing capability is the ability to load different note files and place them on the drawing. The default note file "DEFAULT.NOT" is a generic set of notes which will rarely be useful as is. However, the user can modify the note files (they are text files which any word processor or text editor can modify) and store them under different names. For example, you may want to have a .NOT file for each type of optic you design, and then load the most appropriate note file when the element drawing is generated. The .NOT files must be placed in the <data>\Miscellaneous folder (see "Folders").

If the first line of the note files starts with the number 1, then Zemax will print the note file as provided, with no default notes. If the note file starts with any other symbol, the first default note will be "1) All dimensions in millimeters" or

whatever the current lens units are. To use the default note, start the user supplied notes at note number 2. The line breaks and spacings in the note file will be replicated exactly on the element drawing.

Whenever a new element drawing is generated, or the "Reset" button is pressed, the default settings will be regenerated. The default tolerances are taken from the tolerance data editor. The maximum of the min/max tolerance range is used as the default. For example, if the TTHI thickness tolerance is -.03, +.05, the tolerance value will be 0.05. Only TTHI, TRAD, and TIRR tolerances are considered. If a suitable default cannot be generated, the tolerance is set to zero. Note all tolerance fields are text; and may be edited to suit any requirement.

A handy conversion between radius tolerance and the power tolerance in fringes for a Newton's rings type optical test against a test plate is given by

$$\# \, fringes = \frac{\Delta R}{\lambda} \frac{\rho^2}{R^2}$$

where ΔR is the radius error, λ is the test wavelength, ρ is the radial aperture, and R is the radius of curvature. This formula is an approximation for shallow curvatures. For more information, see Malacara, Optical Shop Testing, J. Wiley & Sons, Inc.

Special characters There are useful special characters that may be inserted and edited in the text fields of the tolerances. The symbols are inserted by pressing and holding Alt on the keyboard and typing a 4 digit number on the numeric keypad. Some common characters are ± - Alt 0177, µ - Alt 0181, ® - Alt 0174, © - Alt 0169.

ISO Element Drawing

This feature creates an ISO 10110 type drawing of surface, singlet, or doublet elements suitable for use in optical shop fabrication.

First Surface The first surface of the element to be drawn.

Show As Choose surface, singlet, or doublet.

Other settings There are many other settings available for this feature that correspond to the tolerances defined in the ISO 10110 specification. See the discussion for details.

The ISO 10110 Element Drawing is an interpretation of the drawing specification "ISO 10110 Optics and Optical Instruments -- Preparation of drawings for optical elements and systems: A User's Guide", by Ronald K. Kimmel and Robert E. Parks, eds., published by the Optical Society of America. For more information see OSA's web site at www.osa.org.

Although the ISO specification only covers the case of single elements, the Zemax ISO drawing also supports surfaces and doublets. However, not all the ISO drawing particulars are incorporated on a drawing of a cemented doublet. If the doublet drawing results are not satisfactory, draw the doublet as two separate singlet drawings instead. This feature does not consider the "Ignore Surface" setting and will draw all requested surfaces, whether they are ignored or not.

Summary of ISO 10110 symbols and codes Zemax does not automatically include default values for all ISO tolerances; however text fields are provided for user specification of these tolerances. The text fields have limited width, and if very long strings are placed in the text fields the text may overrun the columns on the drawing. The following table summarizes the ISO 10110 symbols and codes used by Zemax.

Summary of ISO 10110 symbols and drawing codes

Symbol or Code Number	Description
R	Radius of curvature in lens units. CC indicates concave, CX indicates convex.
Ø	Diameter in lens units.
Øe	Effective diameter in lens units.
n	refractive index
v	Abbe number
(λ)	Coating specification
P; Rq; Lmin, Lmax	Indicates a polished surface on the lens drawing. The details of the polish specification are not provided on the drawing, but are specified in the surface data. Rq = maximum permissible rms surface roughness in micrometers, Lmin/Lmax = minimum and maximum length of sampling interval for rms calculation in mm.
0/A	Stress Birefringence; A = maximum optical path difference in nm/cm.
1/NxA	Bubbles & Inclusions; N = number of bubbles and/or inclusions, A = bubble grade number (size)
2/A; B	Inhomogeneity and Striae; A = homogeneity class number, B = striae class number
3/A(B/C) or 3/A(B/C) RMSx < D or 3/- RMSx < D where x is either t, i, or a	Surface Form Tolerance; A = maximum sagitta error in fringes or - if tolerance is part of radius tolerance, B = peak-to-valley irregularity in fringes or - if no tolerance is given, C = non-spherical rotationally symmetric error in fringes. If (B/C) is replaced by A(B), no tolerance is given. For RMS tolerances, D = maximum rms error in fringes, t = total rms deviation from nominal, i = rms irregularity, a = rms asymmetry remaining after asphericity is subtracted from irregularity.
4/S(L)	Centering Tolerance; S = surface tilt angle in minutes of arc, L = lateral displacement
5/NxA; CN'xA'; LN" x A"; EA'''	Surface Imperfection Method 1; N = number of imperfections, A = grade number (square root of imperfection area), C = coating imperfection designation, N' = number of coating imperfections, A' = grade number, E

	= edge chips designation, A''' = chip protrusion from edge, L = long scratch designation, N" = number of scratches, A" = grade number (scratch width (mm))
5/TV or /TV; EA" 5/RV or RV; EA"	Surface Imperfection Method 2; T = transmissive test, V = visibility class number, R = reflective test, E,A" = edge chips, as in Method 1 above.
6/Hth;L;pdg;fp;nts x np or 6/Eth;L;nts	Laser Irradiation Damage Threshold; Hth = energy density threshold, L = laser wavelength (nm), pdg = pulse duration group, fp = pulse repetition rate (Hz), nts = number of test sites, np = number of pulses per test site, Eth = power density threshold in watt-cm^2.

Diameters and bevels The semi-diameter value is used to determine the drawn radial aperture of each surface. A flat bevel is added to any surface whose semi-diameter is less than the largest semi-diameter being drawn. To place flat bevels on all surfaces, use the "Dia (flat) optional" setting available on the "L Surf Codes 3-4" tab. This optional value must exceed twice the semi-diameter of every surface drawn, otherwise the value is ignored.

Tolerance data When the "Reset from TDE" button is pressed, radius, thickness, index, surface tilt, and surface decenter tolerances are taken from the tolerance data editor. The tolerances are automatically reset after changing the surface number or show as settings. Only TRAD, TCUR, TFRN, TCON, TTHI, TIND, TABB, TSTX, TSTY, TSDX, TSDY, TIRR, TEZI, and TWAV tolerances are considered. If a tolerance is not provided, the tolerance is set to zero. The TFRN tolerance is applied to the surface form tolerance code 3/A. Note all tolerance fields may be edited to suit any requirement.

Sag Table

This feature lists the sag (z-coordinate) of the selected surface at various Y-coordinate distances from the vertex. The best fit spherical (BFS) radius is computed, and the sag of the BFS and the difference between the BFS and the surface is also tabulated. The depth of material that would need to be removed from the BFS to yield the asphere is also listed. The dimensionless slope (dz/dr) of the actual surface and the BFS are given, and the difference between the two.

Only the Y-coordinate of the surface is considered, therefore this feature should not be used if the surface is non-rotationally symmetric.
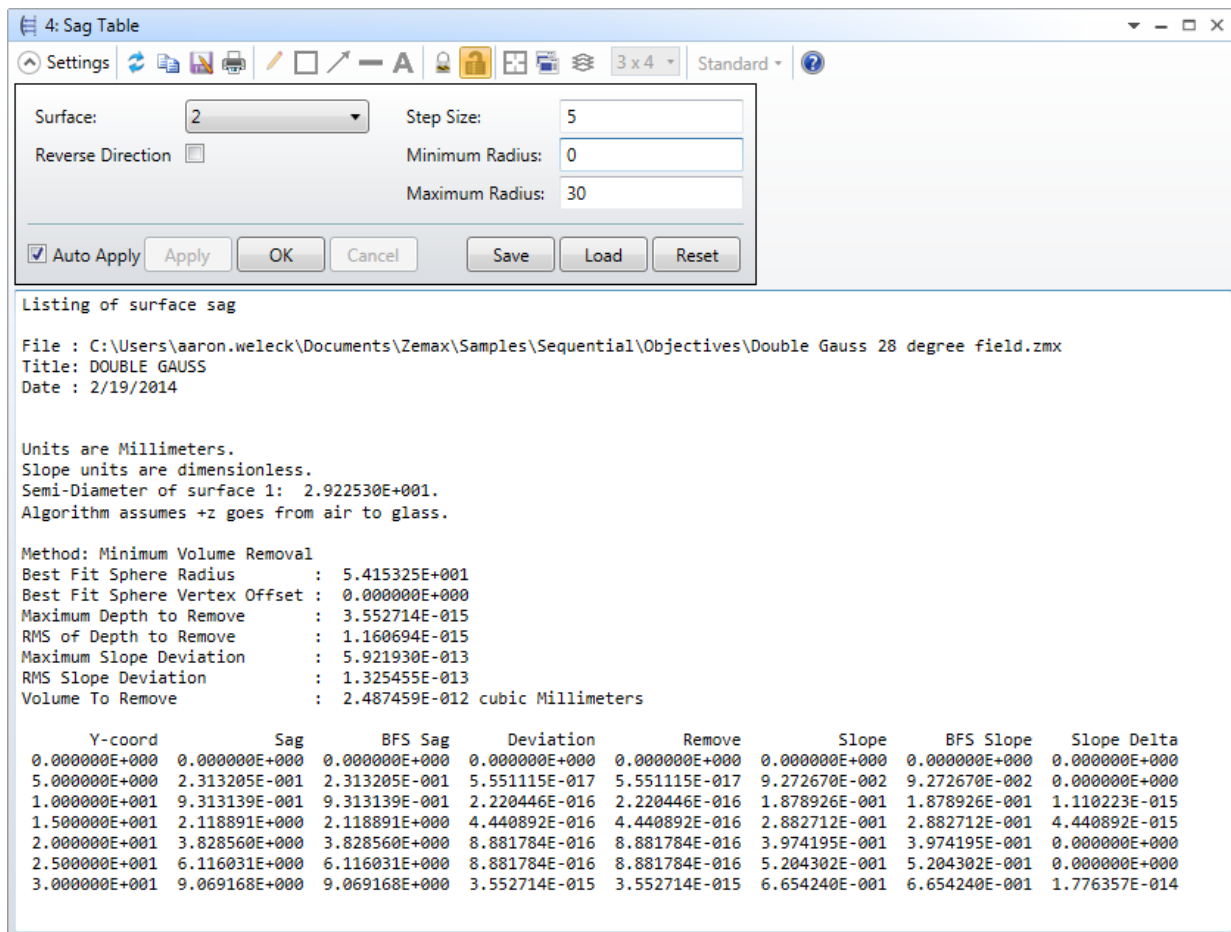
Surface The surface number to use for the calculation.

Step Size The distance between steps measured from the vertex at which the sag is computed. The default value of zero will automatically choose a reasonable step size.

Reverse Direction See "Discussion".

MinR, MaxR The minimum and maximum radial aperture over which the fit is performed. This feature is intended to support radial symmetric annular optics. If both of these values are zero, the MinR is 0.0 and the MaxR is the semi-diameter of the surface.

Discussion

This feature can be used to determine the maximum aspheric deviation of a surface. The sag table is also useful for lens fabrication.

This feature lists out eight columns of data:

- Y-Coord: The y coordinate of the point on the surface being computed.

- Sag: The sag of the surface at the y coordinate.

- BFS Sag: The sag of the best fit sphere. The BFS is determined by the radius of the sphere that minimizes the volume of material that would need to be removed from a spherical surface to yield the aspheric surface.

- Deviation: The difference between the BFS Sag and the Sag, ignoring any vertex offset (see Remove below).

- Remove: The depth of material at the specified y coordinate to remove assuming that the surface was first generated to the best fit sphere. Note that for some aspheric surfaces, material may need to be removed from the

vertex; for other types of aspheres the minimum volume to remove is achieved by removing more material from the central annular zones of the BFS.

- Slope: The slope (dz/dr) of the surface.

- BFS Slope: The slope of the BFS surface.

- Slope Delta: The difference between the Slope and the BFS Slope values.

The BFS radius and position (vertex offset) relative to the aspheric surface depends on the properties of the aspheric surface, and whether the surface defines a boundary between air and glass, or glass and air, and whether the surface is concave or convex. Zemax always attempts to choose a BFS that will "add" material on the air side of the surface. The vertex offset and BFS radius will be selected to minimize the volume of material that needs to be removed from the BFS to yield the asphere surface. For some surfaces, such as those between glass in a cemented doublet, Zemax cannot determine which side is "air". For these surfaces, if the desired data is not generated by default, selecting "Reverse Direction" will choose the other side of the surface as the air surface. The user must check the data and sign convention carefully before any fabrication decisions are made.

The Libraries Tab

This tab provides access to OpticStudio's various libraries and their associated tools.
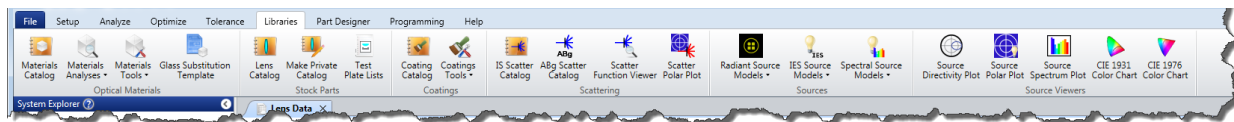
Optical Materials Group

These are the optical materials tools.
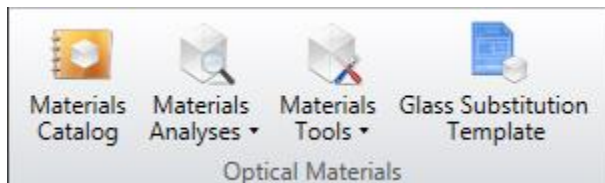
Materials Catalog

Provides access to the glass catalogs.

See "Using Glass Catalogs" in the manual for an extended discussion of this feature.

Materials Analysis

View data on optical materials.
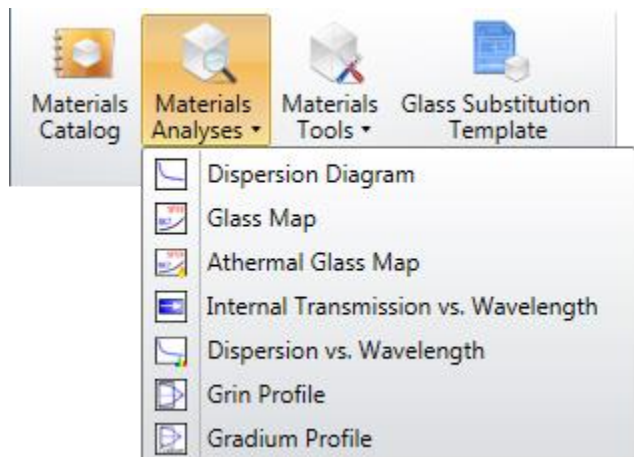
Dispersion Diagram

Plots index of refraction as a function of wavelength for up to 4 different glasses.
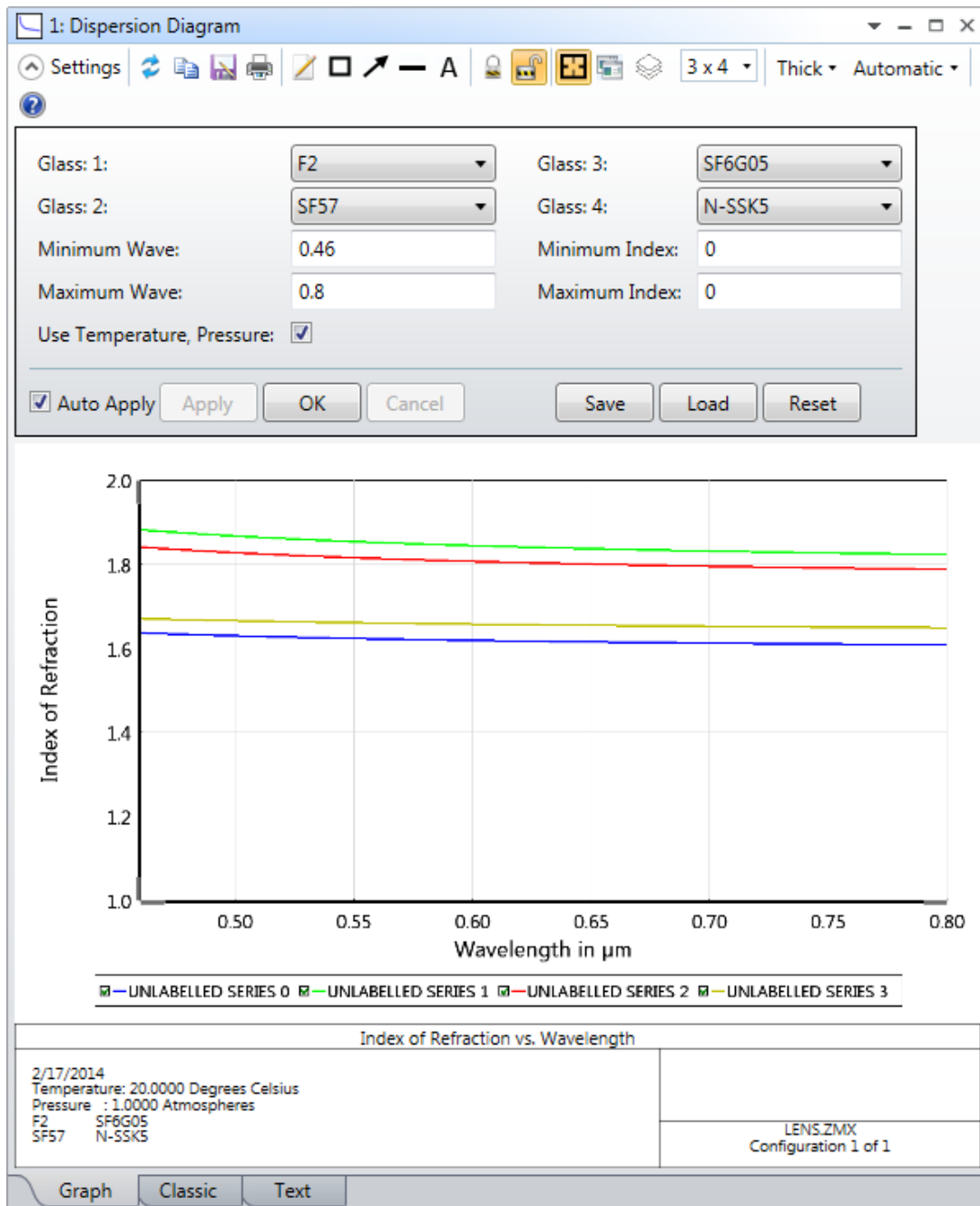
Glass 1 - 4 The name of the glass for which the data is plotted. Up to 4 different glasses may be selected.

Minimum Wave Defines the minimum wavelength for which the data should be plotted.

Maximim Wave Defines the maximum wavelength for which the data should be plotted.

Minimum Index Defines the minimum index for which the data should be plotted. Enter zero for automatic scaling.

Maximum Index Defines the maximum index for which the data should be plotted. Enter zero for automatic scaling.

Use Temperature, Pressure If checked, then the change in index due to temperature and pressure effects will be considered.

Discussion

This feature is useful for comparing the dispersion curves between different glasses. Table glass files (see "Using table glasses") are listed at the bottom of the various "Glass" controls.

If the minimum and maximum wavelength inputs are both zero, then the left X-axis of the plot will be determined by the minimum wavelength for which data are available for any of the input glasses (even if other glasses do not have data at this wavelength) and the right X-axis of the plot will be determined by the maximum wavelength for which data are available for any of the input glasses (even if other glasses do not have data at this wavelength).

Glass Map

Plots names of glasses on map according to the d-light index of refraction and Abbe V-number. The index and Abbe number are computed from the data in the glass catalog. All currently loaded glass catalogs are searched, and glasses within the boundary values specified in the following table are plotted.
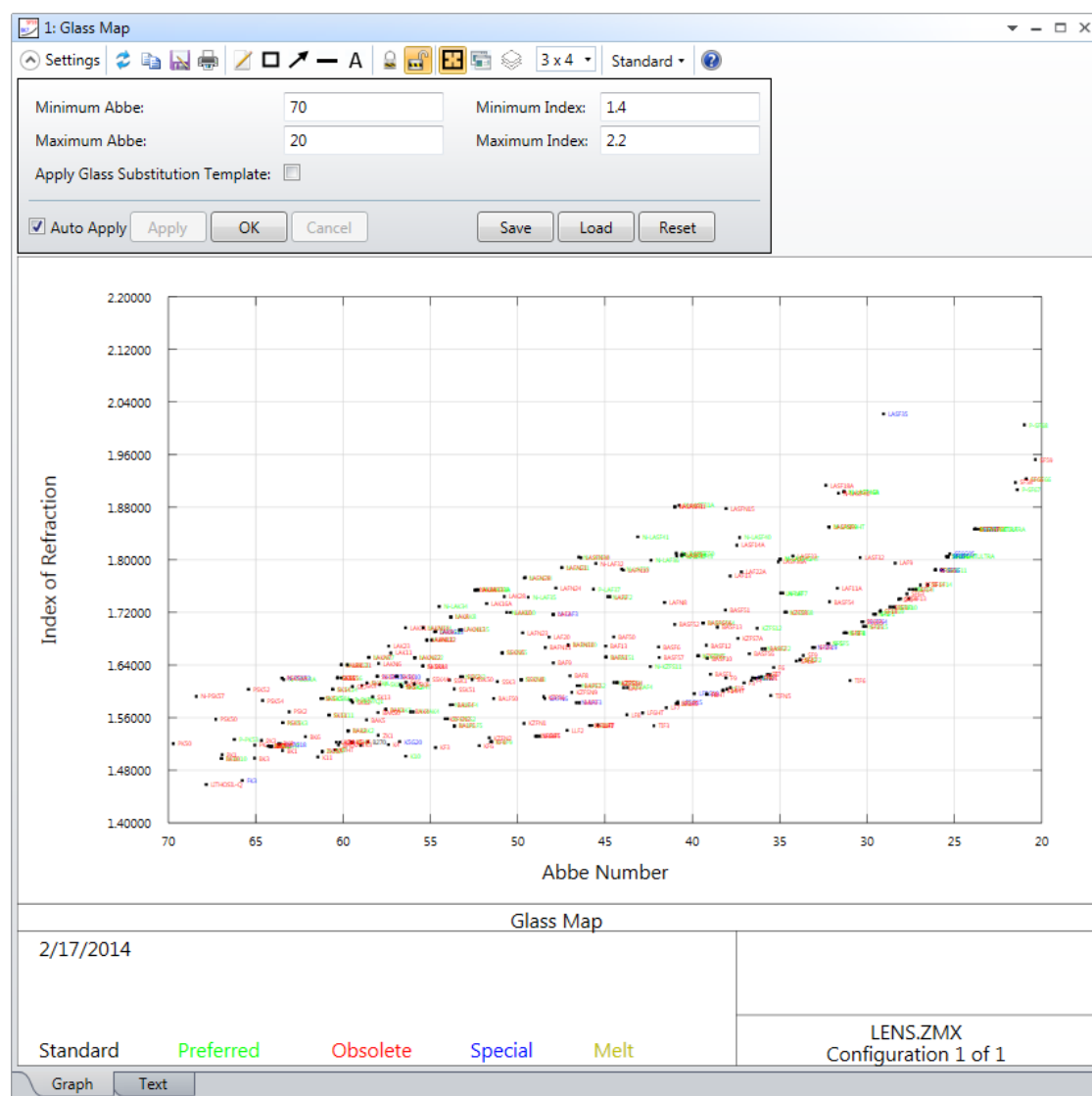
| Min Abbe | Defines the left X- axis of the plot. |
| Max Abbe | Defines the right X- axis of the plot. |

Min Index   Defines the bottom Y- axis of the plot. Enter zero for automatic scaling.

Max Index   Defines the top Y- axis of the plot. Enter zero for automatic scaling.

Apply Glass Substitution Template   If checked, only glasses that meet the glass substitution template will be shown. See "Glass Substitution Template".

Discussion This is useful for locating a glass with particular refractive and dispersive properties. By convention, the glass map is shown with the Abbe number decreasing while going from left to right, which is why the default min and max Abbe numbers seem to be reversed.

Athermal Glass Map

Plots names of glasses on a map according to the chromatic and thermal power. The chromatic and thermal power are computed based upon the currently defined wavelengths and the data in the glass catalog. All currently loaded glass catalogs are searched, and glasses within the boundary values specified in the following table are plotted.
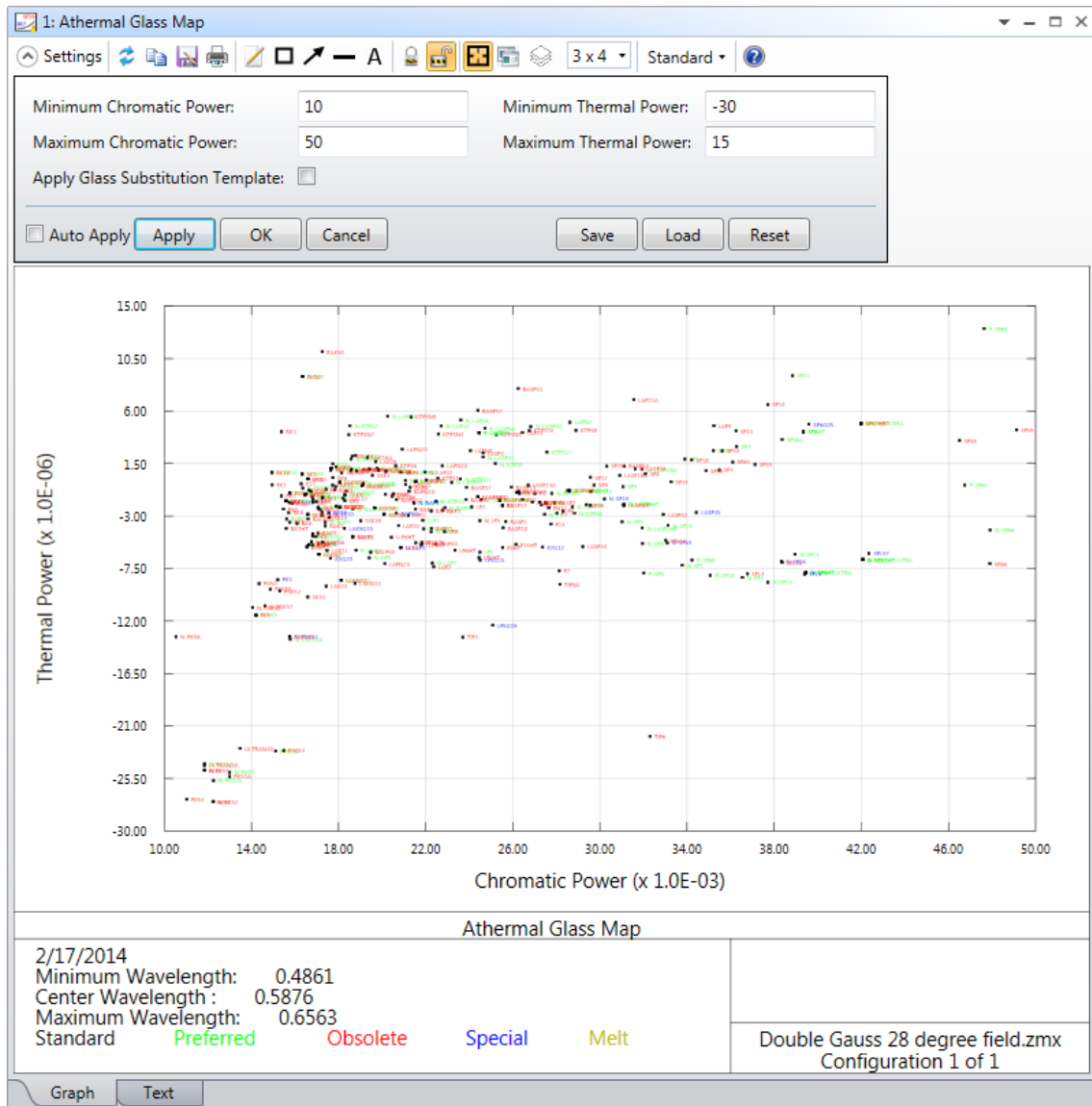
Min/Max Chrom. Power Minimum/Maximum Chromatic power to plot. If both the Min and Max Chromatic power are zero, a default scaling is selected.

Min/Max Therm. Power Minimum/Maximum Thermal power to plot. If both the Min and Max Thermal power are zero, a default scaling is selected.

Apply Glass Substitution Template If checked, only glasses that meet the glass substitution template will be shown. See "Glass Substitution Template".

Reference Glass If a glass is selected, a reference line is drawn from the origin of the plot to the selected glass. Glasses along this line may be used to make athermalized achromatic doublets. See the discussion below.

Discussion

The Chromatic power, which is the inverse of a generalized Abbe V-number, is given by

$$\varphi = \frac{n_{min} - n_{max}}{n_{cen} - 1}$$

where the subscripts on the index values refer to the relative refractive index measured at the minimum, center, and maximum wavelengths, respectively. The wavelength values are determined by the defined wavelengths.

The Thermal power is given by

$$\psi = \frac{\frac{\delta n}{\delta t}}{n_{cen} - 1} - \alpha$$

Where $\frac{\delta n}{\delta t}$ is the derivative of the relative refractive index with respect to temperature measured at the center wavelength and the reference temperature of the glass, and α is the linear expansion coefficient of the glass.

All these values are computed using the data for the glass defined in the glass catalog. The air pressure is assumed to be 1.0.

To make a doublet that is both achromatic and athermalized requires the solution of three simultaneous equations:

$$\Phi = \Phi_1 + \Phi_2$$

$$\Delta\Phi = \varphi_1 \Phi_1 + \varphi_2 \Phi_2$$

$$\frac{d\Phi}{dT} = \psi_1 \Phi_1 + \psi_2 \Phi_2$$

To solve these three equations simultaneously with two glasses requires selection of glasses that lie along a line that also crosses the origin of the athermal glass map. This reference line may be drawn on the map by selecting a reference glass, which defines one point on the line (the other point is the origin of the map).

Internal Transmittance vs. Wavelength

Plots the internal transmittance for any thickness as a function of wavelength for up to 4 different glasses.
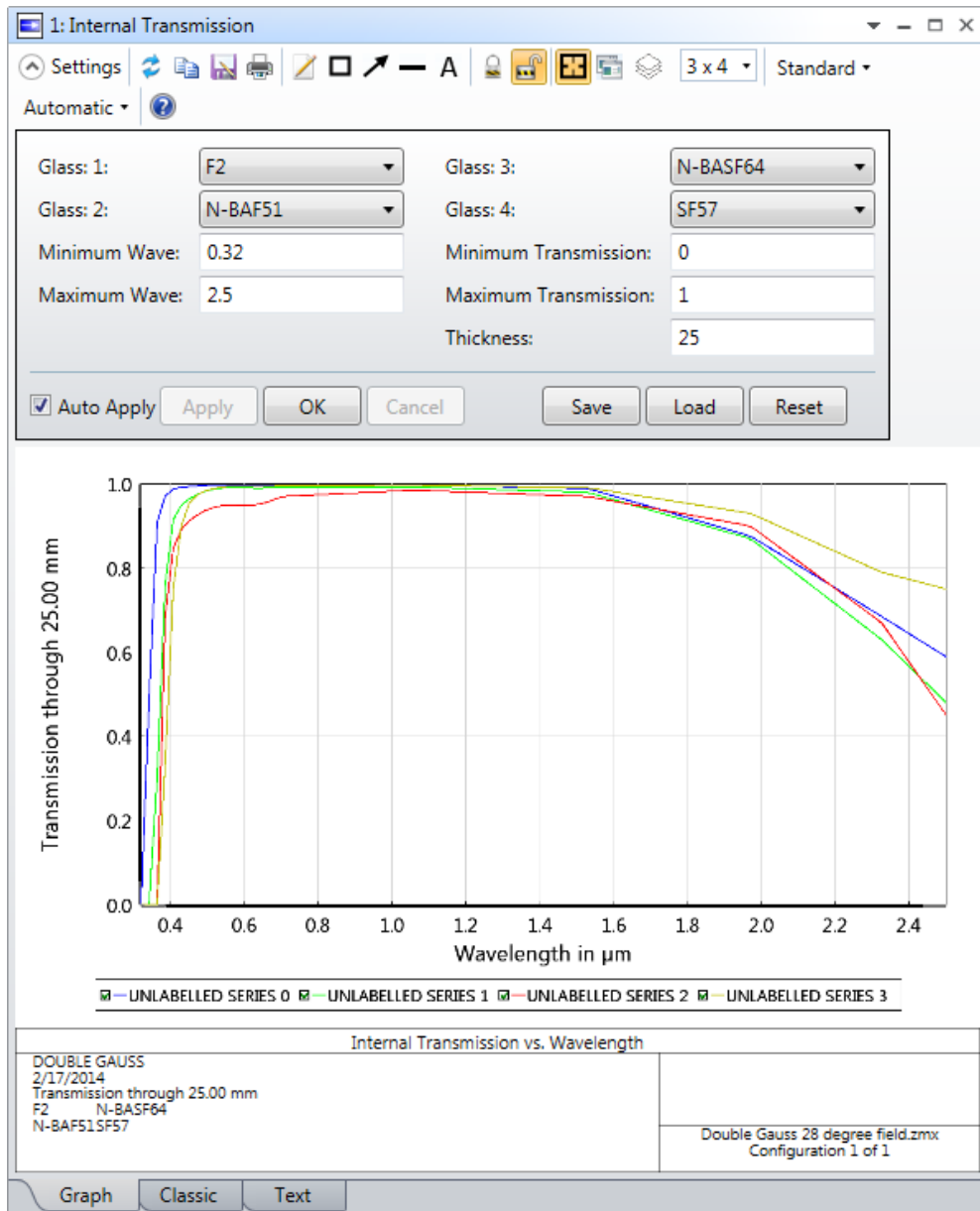
Glass 1 - 4 The name of the glass for which the data is plotted. Up to 4 different glasses may be selected.

Minimum Wave Defines the minimum wavelength for which the data should be plotted.

Maximim Wave Defines the maximum wavelength for which the data should be plotted.

Minimum Transmission Defines the minimum transmittance for which the data should be plotted.

Maximum Transmission Defines the maximum transmittance for which the data should be plotted.

Thickness The thickness of the glasses in millimeters.

Discussion This feature is useful for comparing the transmittance curves between different glasses. Table glass files (see "Using table glasses") are listed at the bottom of the various "Glass" controls. See also "Defining Transmission Data" for more details.

If the minimum and maximum wavelength inputs are both zero, then the left X-axis of the plot will be determined by the minimum wavelength for which data are available for any of the input glasses (even if other glasses do not have data at this wavelength) and the right X-axis of the plot will be determined by the maximum wavelength for which data are available for any of the input glasses (even if other glasses do not have data at this wavelength).

Dispersion vs. Wavelength

Plots the dispersion ($\frac{dn}{d\lambda}$) as a function of wavelength for up to 4 different glasses.
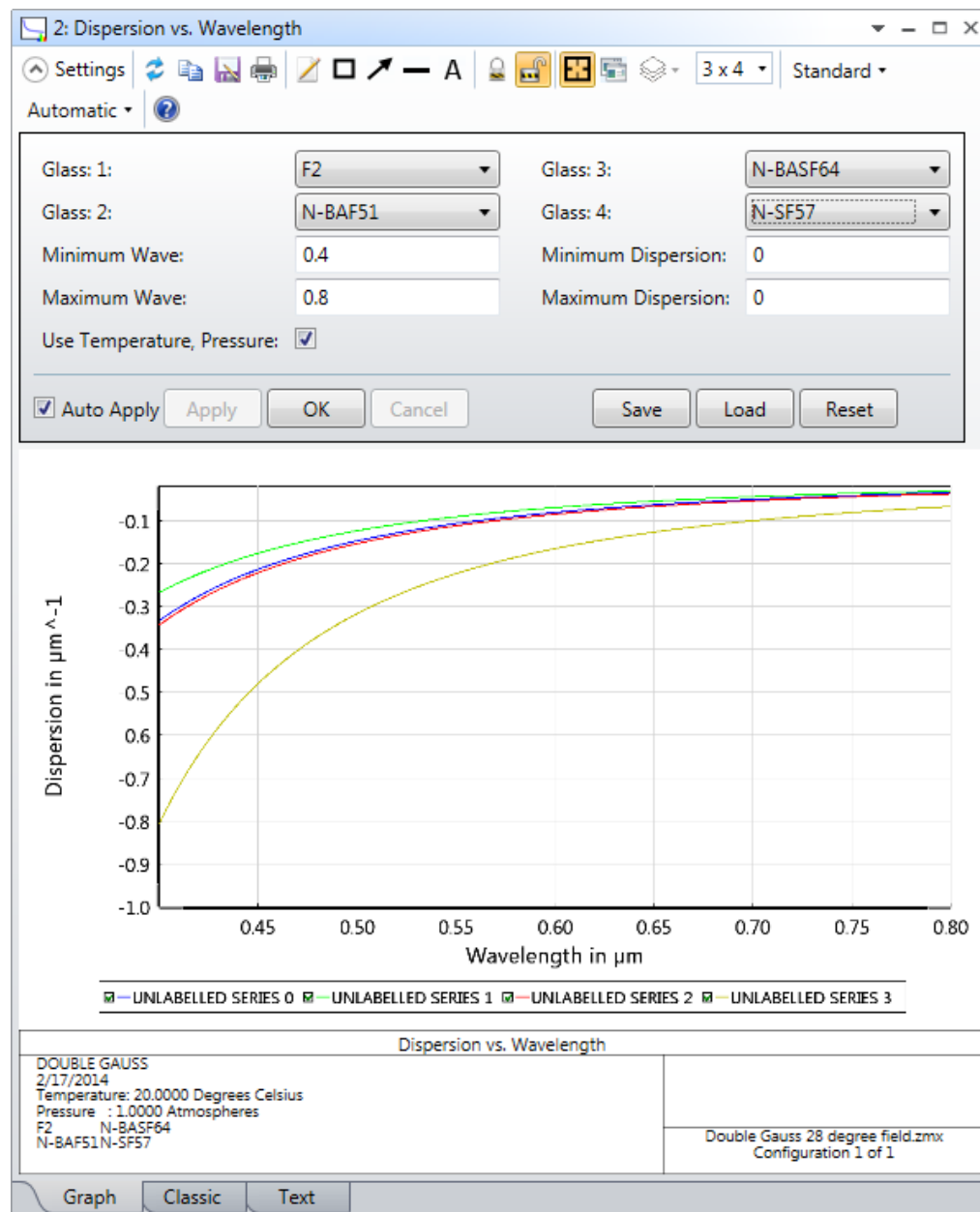
Glass 1 - 4   The name of the glass for which the data is plotted. Up to 4 different glasses may be selected.

Minimum Wave Defines the minimum wavelength for which the data should be plotted.

Maximim Wave Defines the maximum wavelength for which the data should be plotted.

Minimum Dispersion Defines the minimum dispersion for which the data should be plotted. Enter zero for automatic scaling.

Maximum Dispersion Defines the maximum dispersion for which the data should be plotted. Enter zero for automatic scaling.

Use Temperature, Pressure If checked, then the change in index due to temperature and pressure effects will be considered.

Discussion

This feature is useful for comparing how dispersion varies with wavelength for different glasses. Table glass files (see "Using table glasses") are listed at the bottom of the various "Glass" controls.

If the minimum and maximum wavelength inputs are both zero, then the left X-axis of the plot will be determined by the minimum wavelength for which data are available for any of the input glasses (even if other glasses do not have data at this wavelength) and the right X-axis of the plot will be determined by the maximum wavelength for which data are available for any of the input glasses (even if other glasses do not have data at this wavelength).

Grin Profile

Plots the index of refraction along any axis for gradient index surfaces.
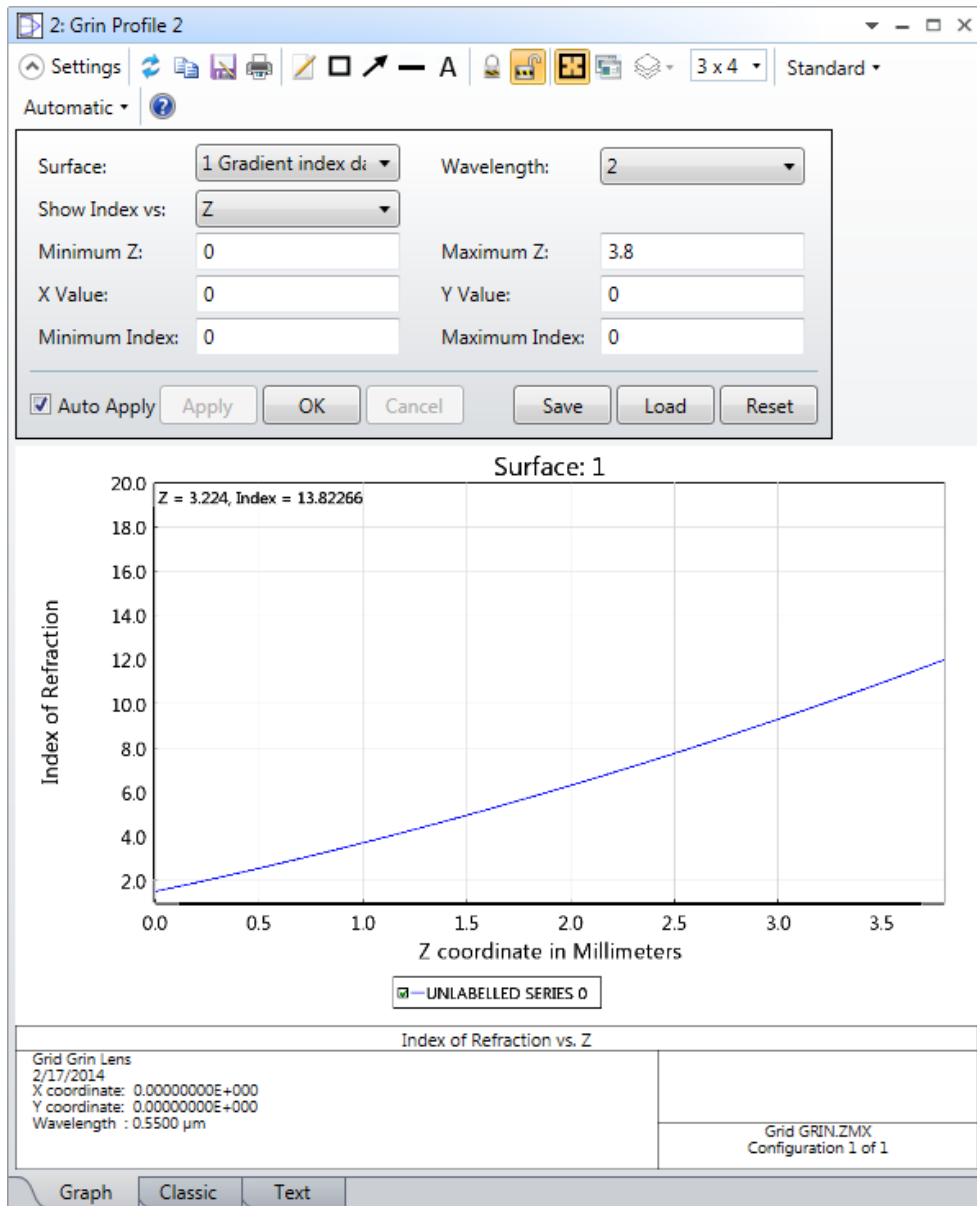
Surface The surface number of the gradient index surface.

Wavelength The wavelength number to use.

Show Index vs. Choose X, Y, or Z as the independent variable.

Min X, Y, or Z Defines the left X- axis of the plot, the minimum value of the independent variable.

Max X, Y, or Z Defines the right X- axis of the plot, the maximum value of the independent variable.

X, Y, or Z Value The value of the two fixed coordinates.

Minimum Index Defines the bottom Y- axis of the plot. Use zero for the default value.

Maximum Index Defines the top Y- axis of the plot. Enter zero for the default value.

Discussion This plot shows index of refraction along either the X, Y, or Z axis for gradient index surfaces.

Gradium™ Profile

Plots the axial index profile of a Gradium surface.
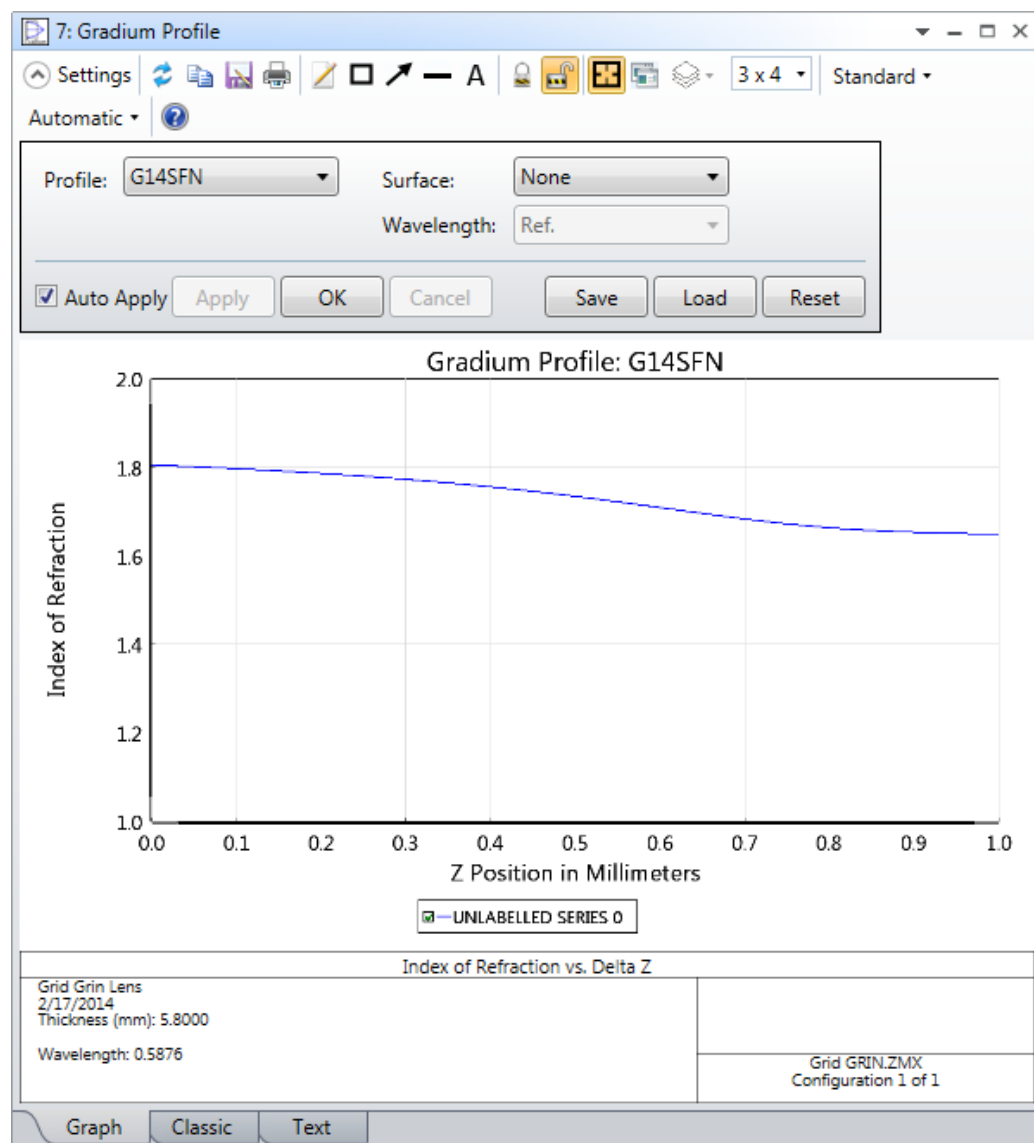
Profile   The name of the profile. See the chapter on surface types for details.

Surface The surface number to plot.

Wavelength   The wavelength number to use for the calculation.

Discussion   If the surface is selected to "None", then the reference wavelength for the profile glass family is used, no matter what wavelength is selected. If a Gradium surface is selected, then any defined wavelength, or the reference wavelength may be selected. Also, if a surface number is selected, then the starting and ending points for the glass blank

will be indicated using an "X" on the plot. The starting and ending positions include consideration of the sag of the surface at the defined semi-diameter.

Materials Tools

Glass Compare

Compares 2 or 3 glass catalogs to assist in finding glasses used in visible light with similar index and dispersion characteristics.
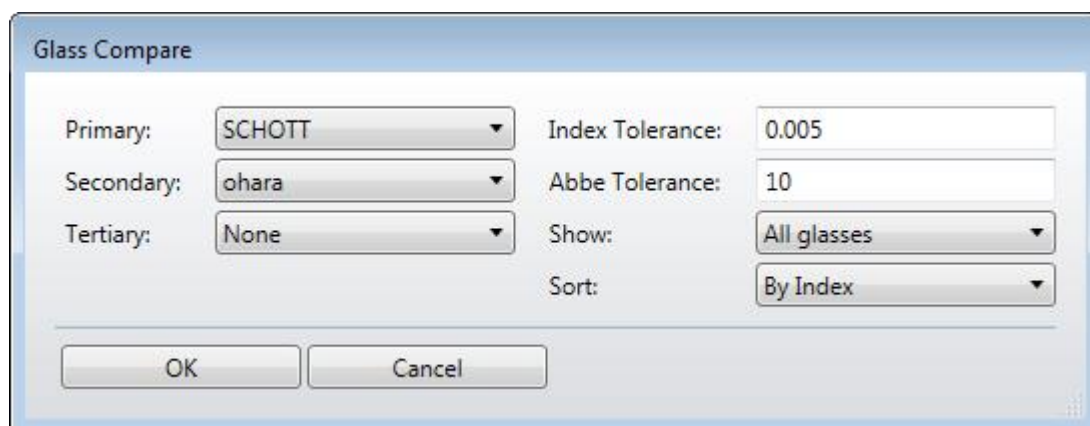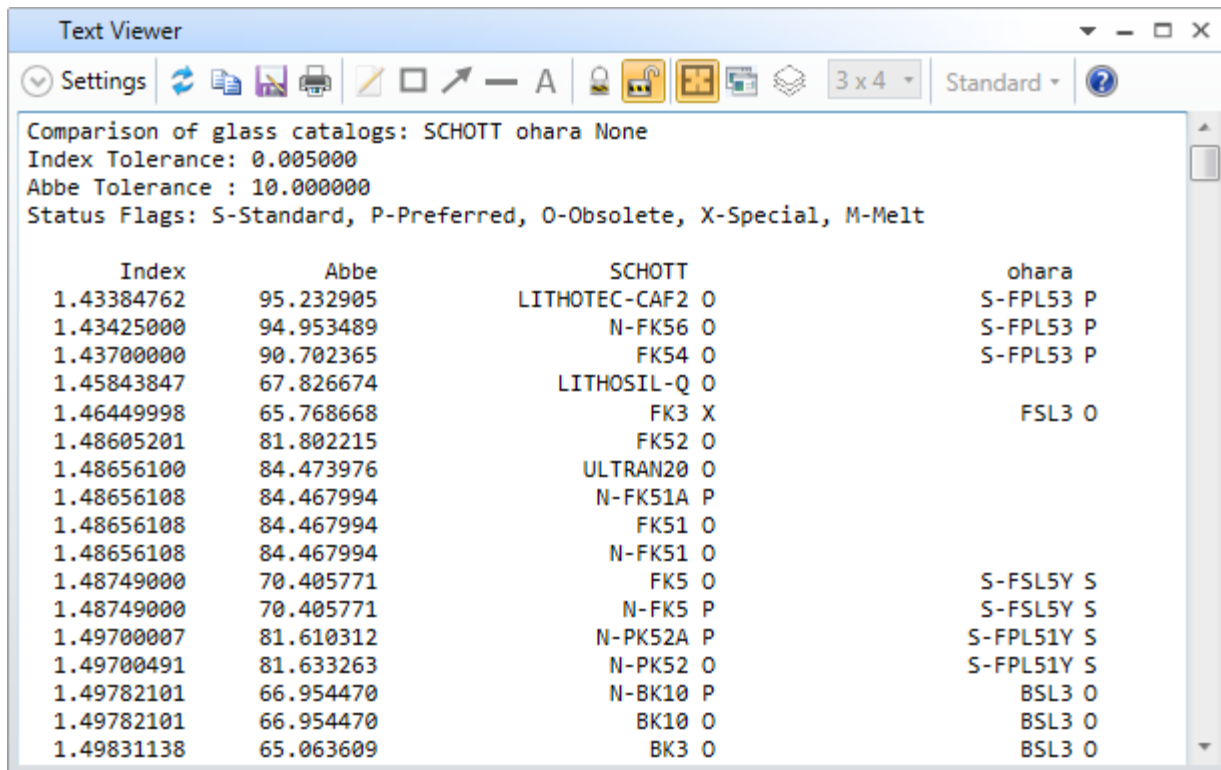
Primary The name of the primary glass catalog. This catalog is used as the reference catalog.

Secondary The name of the secondary glass catalog. Only glasses which are considered equivalent to those in the primary catalog will be listed.

Tertiary The name of the tertiary glass catalog. Only glasses which are equivalent to those in the primary catalog will be listed.

Index Tolerance The maximum allowed difference in index at d-light for a glass from the secondary or tertiary catalog to be considered equivalent to a glass from the primary catalog.

Abbe Tolerance The maximum allowed difference in Abbe number at d-light for a glass from the secondary or tertiary catalog to be considered equivalent to a glass from the primary catalog.

Show Either all glasses in the primary catalog may be listed, or just those used in the current lens file and configuration.

Sort Results may be sorted by index at d-light, name, or Abbe number at d-light.

Discussion

This feature only lists glasses that are used in the F-C wavelength range. If "Show Used Glasses" is selected, the name of the glass displayed in the glass column of the lens data editor is used to determine if the glass from the primary catalog is used by the current lens and configuration. However, there is no check made that the glass from the primary catalog is actually used by the lens. For example, if the current lens uses glass A1 from catalog X, the primary catalog is Y, and Y also contains a glass called A1, then the data for A1 from catalog Y will be displayed, even though the lens actually uses the A1 as defined in catalog X. If more than one catalog uses the same name for different glasses, great care should be taken to verify all data is for the intended glass.

Glass Fitting

Fits glass dispersion data at either a single temperature or thermal dispersion over a range of temperatures.
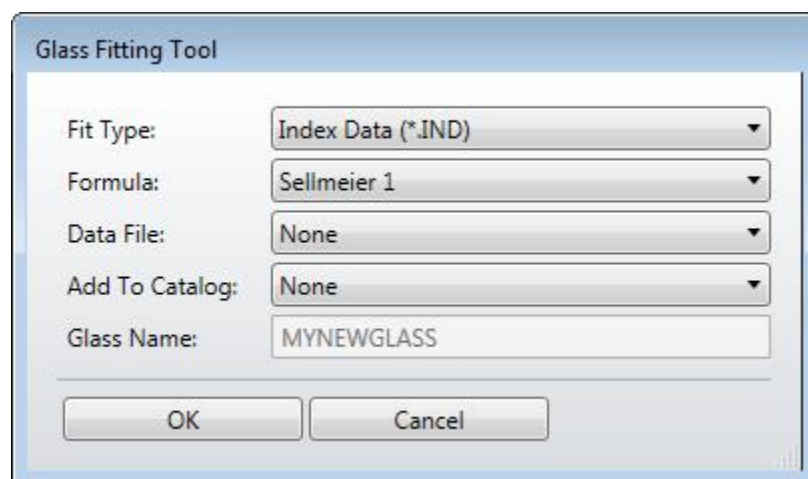
Fit Type   Choose either Index Data or Thermal Index Data.

Formula   Choose the name of the desired glass dispersion formula. For a complete description of the formulas, see "The glass dispersion formulas".

Data File   The name of the data file that contains the dispersion data. When fitting index data, the file extension is IND. For thermal index data, the file extension is TID. See the discussion below for information on the file formats.

Add To Catalog If "none" is selected the data will be fit and a report generated, but the data will not be added to a catalog. If a glass catalog name is selected, then the fit data will be added to the specified catalog. Note that adding glass data to a catalog that is supplied with Zemax is not allowed, therefore only the names of catalogs not supplied with Zemax are listed. If no catalog is listed, then a new catalog must first be created. For information on doing this, see "Creating a new catalog".

Glass Name   The name of the glass to add to the selected glass catalog.

Discussion

There are two different file formats and fitting algorithms supported by this feature. For index data, the text file should end in the extension IND. The first line of the file should contain the word TEMPERATURE followed by the temperature of the glass in degrees Celsius at which the data was measured. The next line should contain the word PRESSURE followed by the ambient air pressure in atmospheres. If one or both of these lines are omitted the default values are 20 degrees and 1.0 atmospheres. The remaining lines consist of two columns of data: wavelength and index. The wavelength values are in micrometers, measured in air at the specified temperature and pressure. The index must be relative to air at the temperature and pressured defined by the first two lines. Zemax will automatically convert the wavelength and index data to be relative to air at a pressure of

1.0 atmospheres at the reference temperature before fitting. A sample file might look like this:

TEMPERATURE 20

PRESSURE 1.0

0.3500000   1.5391663

0.4500000   1.5253195

0.5500000   1.5185224

etc...

For index data, the maximum number of points allowed to be fit is 1000.

For thermal index data, the text file should end in TID. The first line should contain the word PRESSURE followed by the ambient air pressure in atmospheres. If this value is omitted the default value is 1.0 atmosphere. The remaining lines consist of three columns of data: temperature, wavelength, and index. **The temperature of the first data point defines the reference temperature for the glass.** It is important that the reference temperature data be listed first. Data for other temperatures should be listed in subsequent lines. All temperatures are measured in degrees Celsius. The wavelength values are in micrometers, measured in air at the <u>reference temperature</u> and pressure. The index values must be relative to air at the temperature of each data point at the pressure defined by the PRESSURE argument. Zemax will automatically convert the wavelength and index data to be relative to air at a pressure of 1.0 atmospheres before fitting. A sample file might look like this:

PRESSURE 0.0

20.0000000	0.3500000	1.5391663
20.0000000	0.4500000	1.5253195
20.0000000	0.5500000	1.5185224
20.0000000	0.6500000	1.5145203

etc...

-20.0000000	0.3500000	1.5391663
-20.0000000	0.4500000	1.5253195
-20.0000000	0.5500000	1.5185224
-20.0000000	0.6500000	1.5145203

etc...

40.0000000 0.3500000 1.5391663

40.0000000 0.4500000 1.5253195

40.0000000 0.5500000 1.5185224

40.0000000 0.6500000 1.5145203

etc...

The maximum number of points for the reference temperature fitting is 1,000. The total number of points for all temperatures and wavelengths that may be fit is limited only by available memory. Note that the fit will only be as good as the data provided. At least 6 points should be used at each wavelength, and preferably more than 12. At least 6 different temperature values should be provided. Using more than about 30 points at any one temperature tends to only increase the fit time and does not improve accuracy of the fit.

As with all fitting of data, the resulting fit should be carefully checked for suitable accuracy before using the data. It is particularly important to look at the fit index data at temperatures and wavelengths not included in the fitted data, because dispersion formulas can diverge somewhat from reasonable values between data points if the data points are sparse.

Both the IND and TID files should be placed in the folder <glass> (see "Folders" in the user's manual).

Glass Substitution Template

The template is used to define the maximum limits of the glass cost as well as the mechanical resistance properties of glasses. The template is used to select acceptable glasses during global optimization, when converting from model to real glasses (see "Using model glasses" in the user's manual), and when computing the value of the RGLA operand ("RGLA").

Use Glass Substitution Template If checked, the defined parameters are used to restrict the glasses available for substitution by the global optimizer.

Exclude Glasses With Incomplete Data If checked, glasses with incomplete template data will not be selected by the global optimizer. Only data corresponding to non-zero template values are tested before excluding a glass.

Standard, Preferred, Obsolete, Special   These checkboxes indicate the acceptable status settings of glasses. These options are additive. For example, if both Standard and Obsolete are checked, then either Standard or Obsolete glasses are acceptable, Note a single glass may have only one status. See "Description of catalog data".

Maximum Relative Cost The maximum relative cost. The cost of Schott BK7 is 1.0, all other glasses are more expensive, and have larger relative costs. If zero the relative cost is ignored.

Maximum Climatic Resistance (CR)   The maximum allowed value of CR. If zero the CR is ignored.

Maximum Stain Resistance (FR) The maximum allowed value of FR. If zero the FR is ignored.

Maximum Acid Resistance (SR)   The maximum allowed value of SR. If zero the SR is ignored.

Maximum Alkali Resistance (AR)   The maximum allowed value of AR. If zero the AR is ignored.

Maximum Phosphate Resistance (PR)   The maximum allowed value of PR. If zero the PR is ignored.

Save   Saves the current settings to a *.GST file.

Load   Loads the current settings from a previously saved *.GST file.

Save As New Glass Catalog   Saves all the glasses that meets the current criteria into a new glass catalog.

OK   Accepts the current settings.

Cancel Reverts to the prior settings.

Reset   Restores the default settings.

Discussion

If "Use Glass Substitution Template" is unchecked, any glass defined in the current catalogs may be selected if not explicitly excluded (by having "exclude substitution" checked in the glass catalog) and if the wavelength

range is suitable for the current lens. For glasses that do not include the cost or resistance codes, a "?" will be displayed in the glass catalog. If no cost or resistance data is provided for a glass, the glass will not be selected if "Exclude Glasses With No Data" is checked. Template data is stored along with the lens file; so the template parameters are lens file specific. If no glass meets the specified template parameters, the glass substitution status is automatically removed.

The glass substitution template is also used when converting from model to real glasses. When converting model glasses to real glasses, it is possible that no glass can be found that meets the template specification. If this happens, Zemax will first ignore the "Exclude Glasses With No Data" selection and then try again to find a suitable glass. If still no glass can be found that meets the criteria, then Zemax ignores the template entirely and tries again. If still no suitable glass can be found, glass number 1 in the catalog, whatever properties it may have, will be used, even if the glass does not have a suitable wavelength range or is far from the model glass parameters.

Individual tests for Relative Cost, CR, FR, SR, AR, and PR may be activated and deactivated by checking the box next to each description.

Stock Parts Group

These are the stock parts tools.
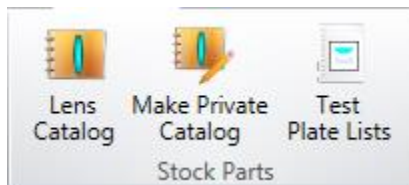
Lens Catalogs

Used to search for a particular lens from or browse through the stock lens catalogs.
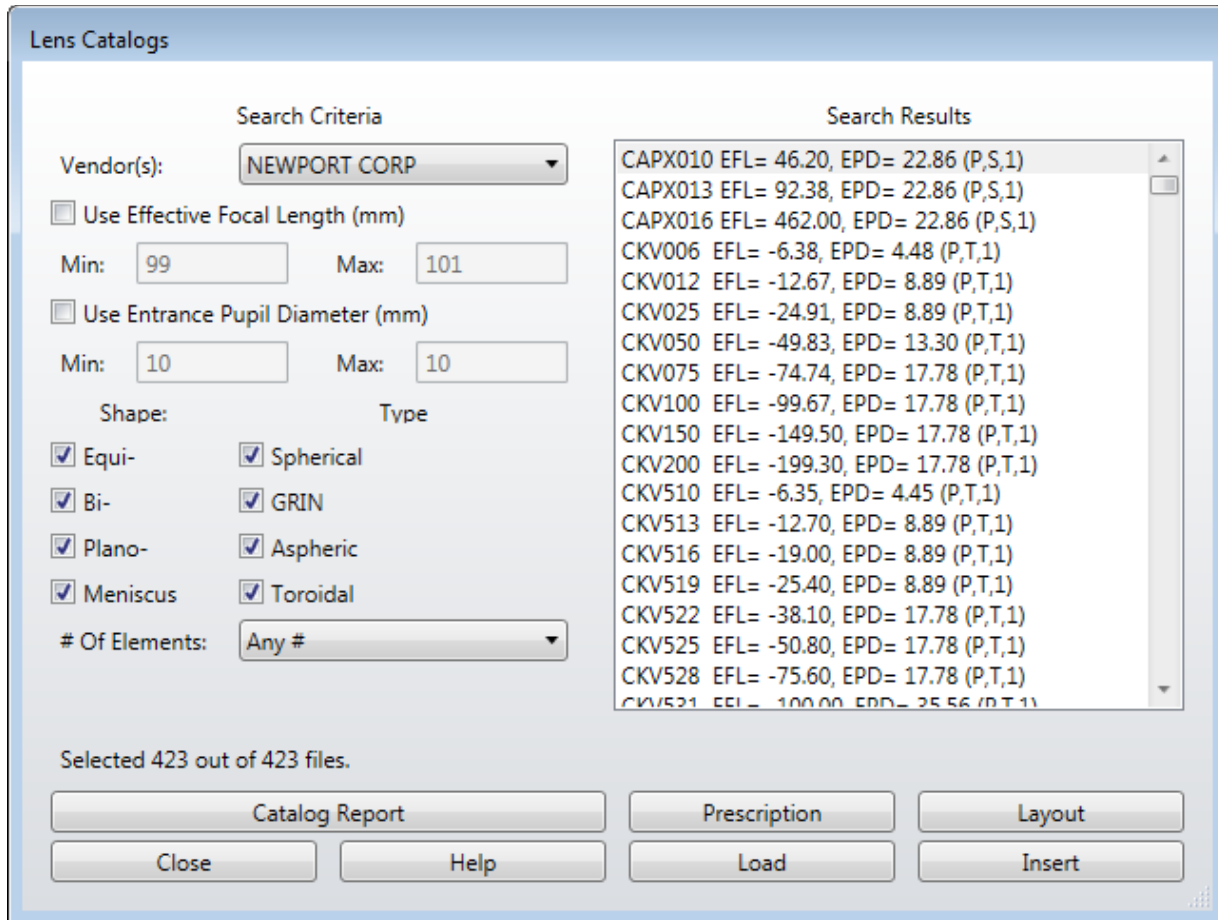
Vendor Lists the available stock lens vendors. Each vendor name is the name of the file which contains the stock lenses available from that vendor. The vendor files must be placed in the <data>\Stockcat folder (see "Folders" in the user's manual).

Use Effective Focal Length If checked, then the specified effective focal length range is used as part of the search criteria, otherwise, any value will be accepted.

Focal Length Min/Max Used to define the range of acceptable effective focal lengths in millimeters.

Use Entrance Pupil Diameter If checked, then the specified entrance pupil diameter range is used as part of the search criteria, otherwise, any value will be accepted.

Diameter Min/Max Used to define the range of acceptable entrance pupil diameters in millimeters.

Equi-, Bi-, Plano-, Meniscus If any of these options are checked, then the search will be limited to lenses meeting at least one of the selected criteria. For example, if both Equi- and Bi- are selected, then both equiconvex/equiconcave and biconvex/biconcave lenses will be included.

Spherical , GRIN, Aspheric, Toroidal If any of these options are checked, then the search will be limited to lenses meeting at least one of the selected criteria. The "Spherical" category is also used as "other" for lenses which are not gradient index, aspheric, or toroidal.

 # Elements Selects the number of elements allowed in lenses that meet the search criteria. Choose "Any #" to include all lenses in the search.

Search Results Lists all lens files in the specified catalog which match the current search criteria.

Search Causes a new search to be conducted for lenses in the specified vendor file based upon the current search criteria.

Load Loads the currently selected lens file into the Lens Data Editor, and updates all windows to display data for the newly loaded lens.

Insert Inserts the currently selected lens file to the existing lens.

In sequential mode, the lens will be inserted into the Lens Data Editor. A prompt will be issued for the insert file options, such as at which surface the lens should be inserted.

In non-sequential mode, the lens will be inserted into the Non-Sequential Components Editor at the bottom of the list of current objects. If the inserted lens consists of more than one component, the second and subsequent component will use the first as its reference object, so only the first component needs to be properly positioned.

Catalog Report Displays a text report of all lenses which meet the search criteria in the currently selected catalog. The values for the criteria used to filter the lenses are displayed at the top of the report. The lenses are then listed

3 times: once sorted alphabetically by part name, once sorted by increasing effective focal length (EFL), and once sorted by increasing entrance pupil diameter (EPD). In each listing, the following data is provided for each lens: part name, EFL, EPD, Shape, Type, and # of Elements. These last three outputs correspond to the 3 part code listed for each lens in the "Search Results" table (see the discussion below for more details).

Prescription Displays a prescription report of the currently selected lens file.

Layout Displays a layout plot (or 3D Layout for systems without axial symmetry) of the currently selected lens file.

Close Closes the dialog box.

Discussion

The "Search Results" table will list the part name, focal length, and diameter of all the lenses from the selected vendor which meet the specified search criteria. After the diameter, a 3 part code is listed, such as (P,S, 1). The first entry is the shape code, which is E, B, P, M, or "?", for Equi-, Bi-, Plano-, Meniscus, or other. Other is used if the lens is multi-element. The second entry is S, G, A, or T, for Spherical, GRIN, Aspheric, or Toroidal. The third entry is the number of elements.

Upon installation, Zemax creates a <data>\Stockcat folder (see "Folders" in the user's manual). Within this folder, many individual files are stored with the extension .ZMF. Each of these files contains a large number of individual ZMX lens files, each of which represent stock lenses available from various vendors. The following table lists the represented vendors and contact data.

STOCK LENS VENDORS

File Name	Website
3M PRECISION OPTICS.ZMF	www.3m.com
ANTERYON.ZMF	www.anteryon.com
ARCHEROPTX.ZMF	www.archeroptx.com
ASPHERICON.ZMF	www.asphericon.com
BEFORT-WETZLAR.ZMF	www.befort-optic.com
COMAR.ZMF	www.comaroptics.com
CVI MELLES GRIOT.ZMF	www.cvimellesgriot.com
DAHENG OPTICS.ZMF	www.dahengoptics.com
DIAS INFRARED.ZMF	www.dias-infrared.de
DIVERSEOPTICS.ZMF	www.diverseoptics.com
EALING.ZMF	www.ealingcatalog.com
EDMUND OPTICS.ZMF	www.edmundoptics.com
EKSMA OPTICS.ZMF	www.eksmaoptics.com
ESCO.ZMF	www.escoproducts.com
GELTECH.ZMF	www.lightpath.com

ISP OPTICS.ZMF	www.ispoptics.com
LIGHTPATH.ZMF	www.lightpath.com
LIMO.ZMF	www.limo.de
LINOS PHOTONICS.ZMF	www.linos-photonics.com
MIDWEST OPTICAL.ZMF	www.midwestopticalsystems.com
NEWPORT CORP.ZMF	www.newport.com
NSG.ZMF	www.nsgamerica.com
OPTICS FOR RESEARCH.ZMF	www.ofr.com
OPTOSIGMA.ZMF	www.optosigma.com
PHILIPS	www.philips.com
QIOPTIC_POLYMER.ZMF	www.qiopticpolymer.com
ROSS OPTICAL.ZMF	www.rossoptical.com
RPO.ZMF	www.rpoptics.com
SIGMA KOKI.ZMF	www.sigma-koki.com
SPECIAL OPTICS.ZMF	www.specialoptics.com
THORLABS.ZMF	www.thorlabs.com

To find a catalog lens that closely matches the properties of one or more surfaces in the Lens Data Editor, place the cursor on the first surface of the surface group in the Lens Data Editor, then select Tools, Lens Catalogs. The default focal length and diameter search ranges will be set to appropriate values for that lens. Zemax computes the lens focal length and diameter and adds plus/minus 5% to define the default search ranges.

The Lens Search tool provides the capability to search through the available lenses to find an appropriate choice. Once selected, the lens file may be loaded or appended to an existing design.

Make Private Lens Catalog

Purpose:

Used to convert a folder full of ZMX files to a stock lens catalog file.
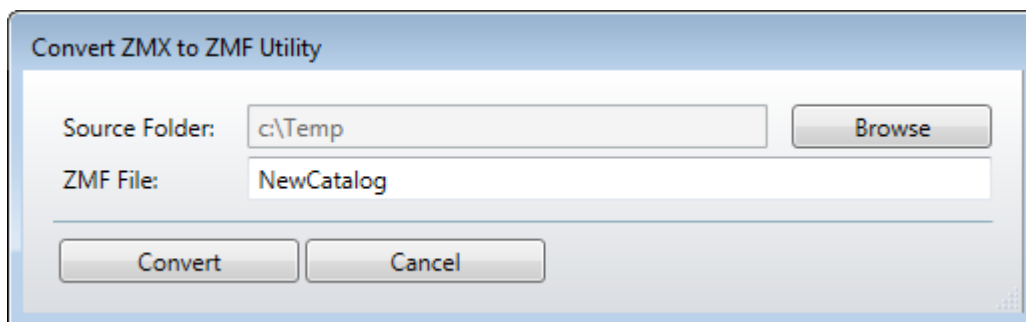
Source Folder Enter the name of the folder where the source ZMX files reside. All ZMX files in this folder will be placed in the ZMF file.

ZMF File Name The name of the ZMF file to be created - this file will be placed in the Source Folder.

Convert Begins the conversion process.

Cancel Closes the dialog box.

Discussion

This feature is used to convert multiple ZMX files into a single ZMF file that can be used by the Lens Catalogs feature. See "Lens Catalogs" for a description of this feature.

There are limitations as to what kind of lenses can sensibly be placed in a private lens catalog. The intended purpose is to include in the catalog only simple lenses that are reasonably described as singlets, doublets, or triplets. Lenses with complex surface shapes, DLLs, non-sequential groups, etc. may not properly convert to the simplified stock lens format used to make these files searchable. File names should use simple ANSI names, as the first space delimited word in the name of the lens file in the source folder is used as the "part name" in the stock lens catalog. For example, a file with the name "Lens number 5.ZMX" will be assigned a part name of "Lens". A better name would be L005.ZMX so that the name "L005" would be used as the part name. Unicode file names are not supported, and will be converted to the nearest equivalent ANSI string, which in some cases may result in odd looking part names.

Test Plate Lists

Displays in a text window a list of test plates from a specific vendor.

```
Listing of test plate file Optimax.tpd

Vendor Data:

  Optimax
  6367 Dean Parkway
  Ontario, NY 14519-8939 USA
  Sales: 877-396-7846
  Tel:   585-265-1020
  Fax:   585-265-1033
  e-mail: sales@optimaxsi.com
  Web: www.optimaxsi.com
  All measurements are in millimeters
  Updated 2 Dec 11


      Plate ID       Radius      Diameter    CC CX
         05503        0.5250        1.4000     X
         03118        0.5270        1.1000     X
         05833        0.5290        1.3000     X
         03116        0.7070        1.6000     X
         05831        0.7860        1.6000     X
         05830        0.7890        1.6000     X
         05907        0.7910        1.4000     X
         05305        0.8170        1.6000     X
         03071        0.8180        1.7000     X
         05312        0.8200        1.6000     X
         03074        0.8200        1.6000     X
         03049        0.8210        1.5000     X
         03113        0.8210        1.8000     X
         03112        0.8220        1.6000     X
```

File Name   The name of the test plate file. The file must be in the <data>\Testplate folder (see "Folders" in the user's manual).

Discussion   All units on the report are in mm. The CC and CX columns indicate the availability of concave and convex test plates, respectively.

Coatings Group

These are the coatings tools.
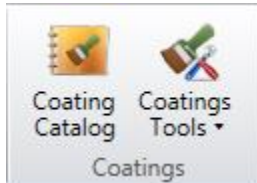
Coating Catalog

This function generates a text listing of the definitions of materials and coatings contained in the coating file for the current lens.
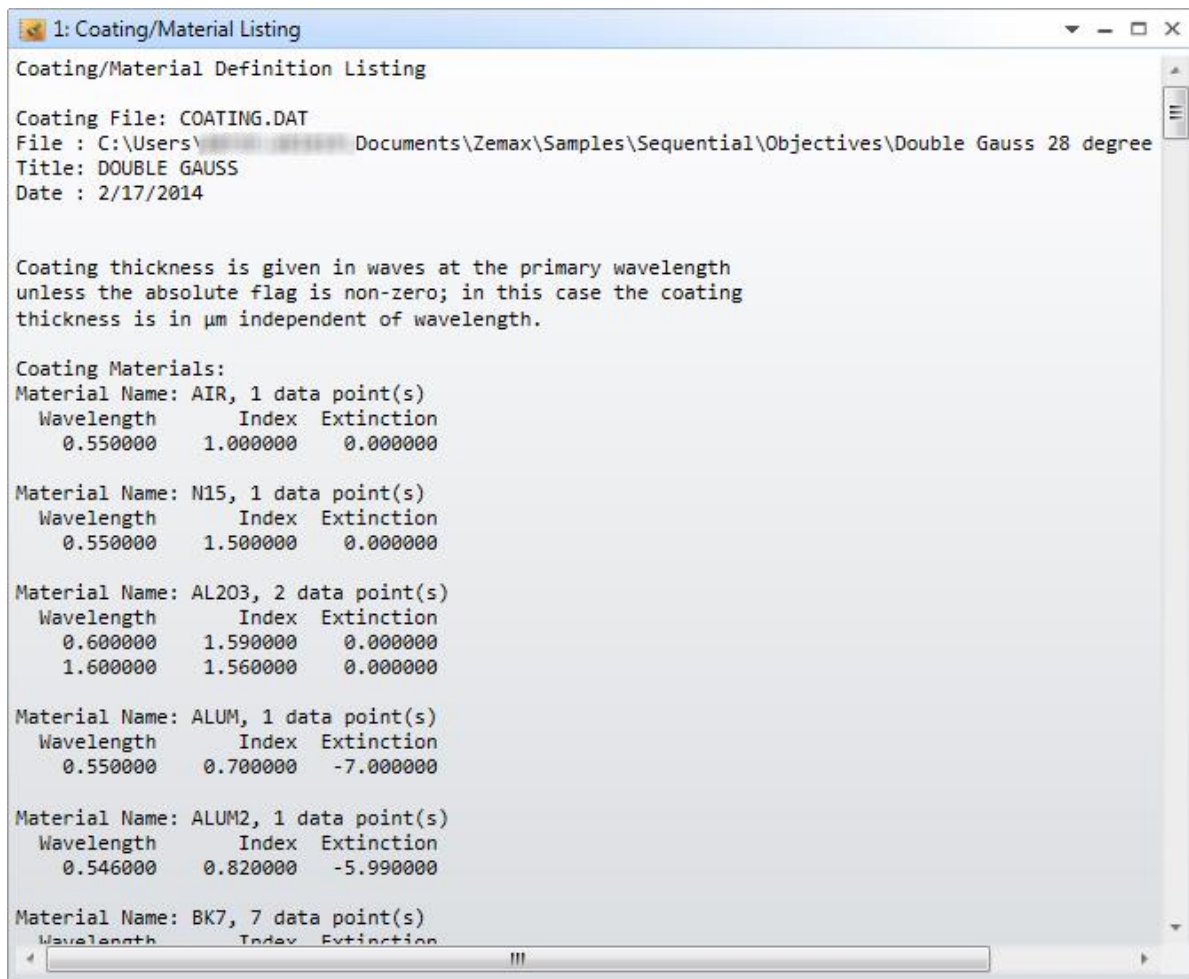
Coating Tools
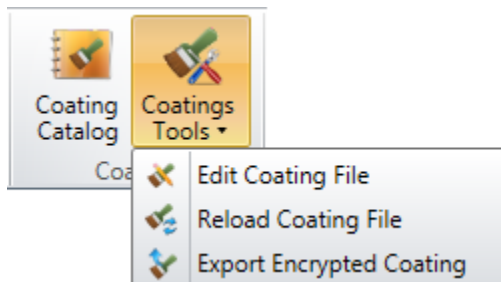
Add and edit coatings; export in an encrypted format.

Edit Coating File

Invokes the Windows NOTEPAD editor to edit the coating file for the current lens. This file contains the material and coating definitions.
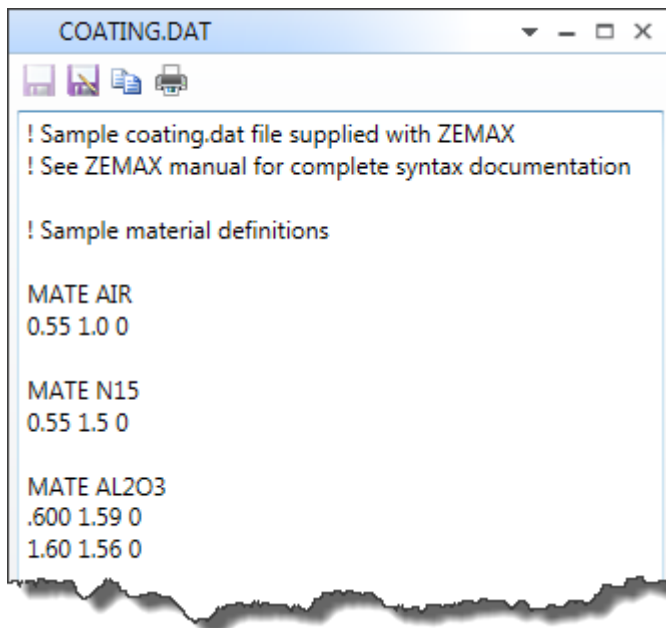
Discussion

See the chapter "Polarization Analysis" in the user's manual. If the coating file is edited, the file must be reloaded manually by selecting "Reload Coating File" as described below, or the file may be saved and reloaded to update the new coating data.

Reload Coating File

Reloads the coating file for the current lens. This is required if the file has been edited since the lens was loaded.

Export Encrypted Coating

This feature exports a single coating to a Zemax Encrypted Coating (ZEC) file. To export an entire lens, see "Export Zemax Black Box Data".
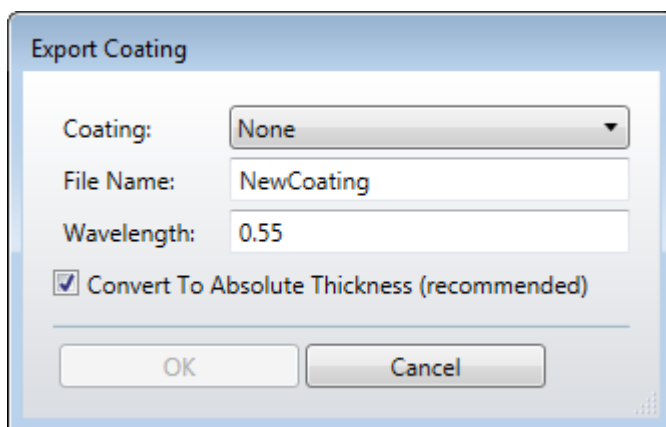
Coating The name of the coating to export.

File Name The name of the file to create. The file name should not include any path or extension. The file will be created in the coating file folder (see "Folders" in the user's manual) and the extension "ZEC" will be automatically appended.

Wavelength The wavelength, in micrometers, measured in air at system temperature and pressure, used to define the layer thicknesses. This value is only used when converting layers from optical thickness to absolute thickness. See below.

Convert to Absolute Thickness Converts optical thickness defined relative to the primary wavelength (for example, a quarter wave thickness) in the coating to absolute thickness. For maximum security, this option should be selected. If the relative optical thickness is left in the coating definition, then the end user of the encrypted coating may be able to determine some properties of the coating by changing the design wavelength. The conversion from thickness in primary waves to absolute thickness is done using the wavelength value from the above setting.

Discussion

Optical coating designers and manufacturers are often hesitant to divulge the details of their coating designs. This feature allows a coating defined in Zemax format to be exported to a single file that is encrypted. The encrypted file may be distributed to users that need to trace rays and perform analysis on the coating as applied to a specific optical system, without the details of the coating design being revealed.

The coating data that is exported includes all the material, index, extinction, dispersion, and layer data required to exactly reproduce the coating performance. No approximations or tables are used. However, no coating tapers, ideal coatings, or table coatings are supported. A maximum of 12 different materials and 400 layers may be used in defining the coating to be exported.

To use an encrypted coating file, see "The ENCRYPTED data section" in the user's manual.

The encryption is based upon a 256-bit algorithm that provides good, but not unbreakable security. This feature is provided as is and the user should determine if the encryption implemented provides adequate security for the intended application before distributing sensitive data in the encrypted format. Radiant Zemax LLC provides no warranty and assumes no liability for the use of this feature.

Scattering Group

These features are used to display data for scatter models.
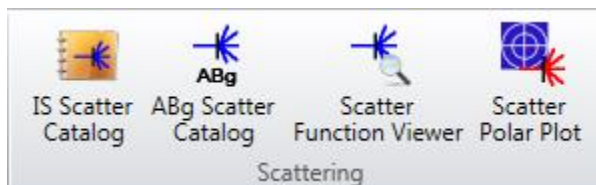
IS Scatter Catalog Data

Browses the online Imaging Sphere library, and downloads Imaging Sphere files.

This feature is only available in the Premium edition of Zemax.

Manufacturer   Select either all or a specific material manufacturer.

Measurement Lamp   Select either all or a specific lamp used during material illumination.

File   Selects the Imaging Sphere scatter file. The name presented is a brief description of the material and manufacturer; the actual file name is displayed below this control.

Discussion

This feature provides access to the online library of Imaging Sphere (IS) scatter data. Scatter data for a number of materials from various manufacturers are available for download and subsequent use in both sequential and non-sequential mode. For information on how the downloaded data are used in Zemax, see the IS Scatter Catalog scattering section under the Technical Reference part of the Help Files.

The catalog of available scatter data profiles is maintained on the local machine. To update this catalog, press the "Download Scatter Catalog" button. This will download the latest catalog data from the online library, and then the available scatter data files will be shown in the dialog box controls.

Descriptive data about the scatter profile, such as whether the scatter measurements were taken in reflection and/or transmission and whether scattering from the material is isotropic or anisotropic, are provided in the dialog box. Additional information regarding material cost and typical usage may be found on the manufacturer's web site.

Once a scatter data file is selected, press "Download Scatter File" to begin the download. Note that IS data files can be large, and may take a long time to download, depending upon the data file size and the connection speed. IS data files are downloaded to the <data>\ScatterData folder (see "Folders" in the user's manual). Currently this folder must be located on the user's local machine; it cannot reside on a network drive.

To close this dialog box, press "Close".

ABg Scatter Data Catalogs

Provides access to the ABg scatter data catalogs.

File The current ABg scatter data file name. This file must be placed in the <data>\ABg_Data folder (see "Folders" in the user's manual); the file name may be selected on the System, General dialog box on the Files tab.

Name The name of the ABg data being edited. This name must be in uppercase.

Wave The wavelength in micrometers at which the ABg data is defined (or was measured). If this value is zero, no scaling with wavelength will be performed. Otherwise, the A and B values will be scaled with wavelength as defined in the discussion below.

Use If checked, then the displayed row will be used. Up to 10 data rows may be defined for each ABg data name.

Angle The angle of incidence in degrees. The value should be between 0.0 and 90.0, inclusive. The BSDF for intermediate angles of incidence are linearly interpolated in cosine space. If data is required at angles less than the smallest angle defined, data for the smallest angle defined will be used. Data for the largest angle defined will be used for any incidence angles greater than the largest angle defined.

A, B, g The A, B, and g values at the specified angle of incidence.

TIS Pressing this button will compute the total integrated scatter (TIS) for the displayed data. This value must be less than 1.0 for accurate scattering results. See the discussion.

Save Saves the modified data to the catalog.

Save As Saves the modified data to a catalog of a different name.

Exit Closes the dialog box.

Insert Inserts a new ABg data entry into the catalog; and prompts for the name of the new entry.

Delete Deletes the displayed ABg data entry.

Reload Loads from the last saved data file the ABg data; eliminating any changes made since the last save.

Sort   Sorts in ascending order by angle the displayed data. The data is automatically sorted whenever the catalog is saved or loaded.

Report Sends the data to a text window which may then be printed.

Rename Prompts for a new name for the ABg data.

Discussion

ABg data values are used to define scattering properties of both sequential and non-sequential surfaces. Each Zemax lens files may use a separate ABg data catalog if desired. The ABg data file used is specified on the System, General dialog box on the Files tab. The selected ABg catalog only may be edited from within Zemax using the ABg Scatter Data Catalog dialog box.

For a theoretical discussion of ABg scattering; see "ABg model scattering" in the user's manual.

The integral of the ABg distribution function over the unit vector circle that represents physically possible scattering ray directions determines the total integrated scatter, or fraction to scatter. This integral is automatically computed by Zemax when required; to view this integral press the TIS button. The TIS must be less than 1.0; otherwise conservation of energy may not be preserved and the scattering will not be correct.

Wavelength scaling of ABg data

If a reference wavelength other than zero is defined, then the A and B coefficients are scaled according to the following relations:

$$A' = A \left(\frac{\lambda}{\lambda_{ref}} \right)^{g-4}$$

and,

$$B' = B \left(\frac{\lambda}{\lambda_{ref}} \right)^{g}$$

where λ is the wavelength being traced and the unsubscripted A, B, and g parameters are measured at the reference wavelength, λ_{ref}.

When using the ABg scatter models, Zemax can only scatter light using ABg parameters scaled for wave- lengths defined in the System Wavelengths Dialog box. If you wish to define a broadband source via a Source Color spectrum (e.g. Tristimulus XYZ), and the ABg data varies strongly with wavelength, then you need to defined a reasonable number of System Wavelengths to span the bandwidth of the source. For an arbitrary wavelength being traced, Zemax will use the closest System Wavelength to determine the ABg coefficients and scatter distribution. This is the only situation in which System Wavelengths need to be defined when using a Source Color model.

Scatter Function Viewer

This feature is only available in the Professional and Premium editions of Zemax.

Displays BSDF as a function of the magnitude of the scatter vector x.

Scatter Model   Selects the scatter model for which to display data.

Wavelength   The wavelength number to use. ABg and ABg File scattering both use the specified wavelength, along with the reference wavelength defined in the ABg catalog, to scale the ABg coefficients. IS Scatter Catalog scattering uses data in the ISX file corresponding to the wavelength which is closest to the specified wavelength.

Fraction   The fraction of total energy to scatter. This value must be between 0 and 1. The fraction to scatter linearly scales the BSDF. This fraction is ignored for ABg and ABg File scatter functions and must be set to 1 for the BSDF and IS Scatter Catalog scatter functions.

Sigma/Angle   For Gaussian scattering: The sigma value determining the width of the Gaussian distribution on the projected plane.

For BSDF and IS Scatter Catalog scattering: The angle (in degrees) describing the orientation of the surface in the system being modeled relative to the orientation of the surface while measured.

F i l e   N a m e / D L L Name   For ABG scattering: The name of the ABg scattering data to use. ABg scatter data names are defined in the ABg Scatter Data Catalogs; see "ABg Scatter Data Catalogs".

For ABg File scattering: The name of the ABGF file to use. See "ABg File scattering" in the user's manual.

For BSDF scattering: The name of the BSDF file to use. See "BSDF scattering" in the user's manual.

For IS Scatter Catalog scattering: The name of the ISX file to use. See "IS ScatterCatalog scattering" in the user's manual.

For User Defined scattering: The name of the user defined scattering DLL to use. See "User defined scattering" in the user's manual.

DLL File Name The name of the data file, if any, used by the user defined scattering DLL.

(parameter names)   These parameters are for user defined scattering DLLs.

Bins   The number of histogram bins to use in the computation and plot of the BSDF for user defined scattering. Using a larger number of bins increases the resolution of the plot, at the expense of requiring more rays to generate smooth data.

Rays   The number of rays to scatter to determine the shape of the BSDF curve for BSDF, IS Scatter Catalog, or user defined scattering.

Alpha X, Y   The angle in the X-Z and Y-Z planes respectively, of the specular ray. See the discussion for details.

Mode   Only used for IS Scatter Catalog scattering. Defines the measurement mode to display from the ISX file (reflection or transmission).

Side   Only used for IS Scatter Catalog scattering. Defines the measurement side to display from the ISX file (front or back).

Sampling   Only used for IS Scatter Catalog scattering. Defines the angular resolution with which to read data from the ISX file (5, 2, or 1 degree in both the radial and azimuthal direction).

X, Y Minimum The minimum value of the scatter vector x, and BSDF (y), to plot, respectively. Note these are powers of 10 as the BSDF plot is a log-log plot.

X, Y Decades   The number of log scale decades to plot in the x and BSDF (y) directions, respectively.

Discussion

This feature plots or lists BSDF data for any of the surface scattering models supported by Zemax. The BSDF plot is a log - log plot, with BSDF on the vertical axis, and the magnitude of the scattering vector x on the horizontal axis. The scattering vector x is a 2D vector that defines the change in the x and y direction cosines between the specular ray and the scattered ray, when the local surface normal is oriented along the z direction. The vector x is therefore the difference between the specular and scattered ray when these rays are projected onto the plane of incidence. For a detailed discussion and drawing illustrating the scattering conventions used by Zemax, see "Scatter models" in the user's manual.

For User Defined, IS Scatter Catalog, BSDF, ABg, and ABg File scattering, the Alpha X and Y values are used to define the specular ray direction cosines. The defining equations which convert angles to cosines are as follows:

$$n = \frac{1}{\sqrt{1 + (\tan \alpha_x)^2 + \left(\tan \alpha_y\right)^2}}$$

$$l = \text{ n} \tan \alpha_x$$

$$m = \text{ n} \tan \alpha_y$$

The comments below apply to each of the available scattering models:

Lambertian: Lambertian scattering is independent of incident angle and wavelength. The total integrated scatter (TIS) is 1.0. The resulting BSDF "curve" is a flat line with a value of $1/\pi$.

Gaussian: Gaussian scattering is symmetric about the specular ray projection. The TIS is always 1.0.

ABg: The ABg values are defined in the ABg Scatter Data Catalogs; see "ABg Scatter Data Catalogs" for more information. The TIS is not generally 1.0; the actual value is listed in the output plot or listing. The ABg data may be scaled with wavelength; if a reference wavelength is defined in the catalog. ABg data is also generally a function of incident angle. Zemax interpolates the BSDF data between the two points with the angles of incidence closest to that of the defined specular ray.

ABg File: All ABg profiles specified in the ABGF file must be defined in the currently loaded ABg Data File (see "ABg Data File" in the user's manual). Each ABg profile is treated as it would be under the ABg scatter model. Thus, ABg data for each profile will be scaled with wavelength if a reference wavelength is defined in the ABg Scatter Data Catalog for that profile (see "ABg Scatter Data Catalogs"), and the ABg data for each profile will be

interpolated between the two points with angles of incidence closest to that of the defined incident angle. The wavelength scaling and interpolation over angle of incidence will be performed for each profile specified in the ABGF file before all profiles are summed. The TIS for the sum of the ABg profiles (whether it be the weighted or absolute sum) is not generally 1.0; the actual value is listed in the output plot.

BSDF: The BSDF values are defined in an input file; see "BSDF scattering" in the user's manual. The TIS is not generally 1.0; the actual value is listed in the output plot. As described in detail in the Knowledge Base, Zemax performs limited interpolation of the input data. Thus, the input data should have sufficient resolution (specifically with regards to the incident, azimuthal, and radial angles) to accurately describe any variations of importance in the BSDF data. To generate the plot for this case, a number of identical specular rays are scattered by the model, and the resulting scatter vector x values are binned in a histogram, normalized, and plotted (same method used for "User Defined" scattering, as described below).

IS Scatter Catalog: The BSDF values are defined in an input file generated by the Radiant Zemax IS-SA device; see "IS Scatter Catalog scattering" in the user's manual. This model is functionally very similar to the BSDF scattering model described above, but with data contained in a high resolution binary file rather than an ASCII text file. The spectral variation of the scattering distribution may also be present in an ISX file. As with the BSDF scattering model, the TIS is not generally 1.0; the actual value is listed in the output plot. To generate the plot for this case, a number of identical specular rays are scattered by the model, and the resulting scatter vector x values are binned in a histogram, normalized, and plotted (same method used for "User Defined" scattering, as described below).

User Defined: Zemax cannot determine the exact curve of the BSDF for user defined scattering. As an alternative, a number of identical specular rays are scattered by the DLL, and the resulting scatter vector x values are binned in a histogram, normalized, and plotted. BSDF plots made in this way are typically noisy in a complex way that depends upon the nature of the scattering function defined by the DLL. Note that if the BSDF as defined by the DLL is not symmetric about the specular ray in the projected tangent plane, then this plot is not an accurate description of the BSDF.

Scatter Polar Plot

This feature is only available in the Professional and Premium editions of Zemax.

Displays BSDF as a function of radial and azimuthal angle on a 2D polar plot.

Scatter Model   Selects the scatter model for which to display data.

Wavelength   The wavelength number to use. ABg and ABg File scattering both use the specified wavelength, along with the reference wavelength defined in the ABg catalog, to scale the ABg coefficients. IS Scatter Catalog scattering uses data in the ISX file corresponding to the wavelength which is closest to the specified wavelength.

Fraction   The fraction of total energy to scatter. This value must be between 0 and 1. The fraction to scatter linearly scales the BSDF. This fraction is ignored for ABg and ABg File scatter functions and must be set to 1 for the BSDF and IS Scatter Catalog scatter functions.

Sigma/Angle   For Gaussian scattering: The sigma value determining the width of the Gaussian distribution on the projected plane.

For BSDF and IS Scatter Catalog scattering: The angle (in degrees) describing the orientation of the surface in the system being modeled relative to the orientation of the surface while measured.

File Name/DLL Name   For ABG scattering: The name of the ABg scattering data to use. ABg scatter data names are defined in the ABg Scatter Data Catalogs; see "ABg Scatter Data Catalogs".

For ABg File scattering: The name of the ABGF file to use. See "ABg File scattering" in the user's manual.

For BSDF scattering: The name of the BSDF file to use. See "BSDF scattering" in the user's manual.

For IS Scatter Catalog scattering: The name of the ISX file to use. See "IS Scatter Catalog scattering" in the user's manual.

For User Defined scattering: The name of the user defined scattering DLL to use. See"User defined scattering" in the user's manual.

DLL File Name   The name of the data file, if any, used by the user defined scattering DLL.

(parameter names)   These parameters are for user defined scattering DLLs.

Incident Angle   The angle of incidence (in degrees) for which the BSDF should be calculated and displayed. The BSDF is a function of incident angle for all scatter models except Lambertian.

Show As   Grey scale or false color displays which are consistent with the color palette used elsewhere in Zemax. For BSDF and IS Scatter Catalog scattering, an IS-SA false color display is available, which uses the same color palette as that employed by the Radiant Zemax IS-SA device. This latter option allows for direct comparison of BSDF data between Zemax and IS-SA.

Plot Scale   If non-zero, the maximum value for the plot will be scaled to this value. If the value is zero, then the maximum value for the plot will be determined from the data.

Mode   Only used for IS Scatter Catalog scattering. Defines the measurement mode to display from the ISX file (reflection or transmission).

Side   Only used for IS Scatter Catalog scattering. Defines the measurement side to display from the ISX file (front or back).

Sampling   Only used for IS Scatter Catalog scattering. Defines the angular resolution with which to read data from the ISX file (5, 2, or 1 degree in both the radial and azimuthal direction).

Rays   The number of rays to scatter to determine the shape of the BSDF curve for user defined scattering.

Radial Bins   The number of bins used to display the data in the radial (polar) direction, covering 180 degrees.

Azimuthal Bins The number of bins used to display the data in the azimuthal direction, covering 360 degrees.

Normalize   If selected, the data will be normalized to the maximum BSDF value.

Discussion

This feature plots the 2D BSDF data for any of the surface scattering models supported by Zemax. The plot is provided in standard polar coordinates, with the radial angle extending outwards from the center and the azimuthal angle moving counter-clockwise from the +X axis. The plot resolution is determined by the number of radial and azimuthal bins specified (for the BSDF scatter model, the resolution of the input file will also play a role

- see more details below).

For a detailed description of each of the available scatter models, see "Scatter models" in the user's manual. The comments below apply to each of the available scatter models:

Lambertian: Lambertian scattering is independent of incident angle and wavelength. The total integrated scatter (TIS) is 1.0. The resulting BSDF plot is therefore uniform.

Gaussian: Gaussian scattering is symmetric about the specular ray projection. The TIS is always 1.0.

ABg: The ABg values are defined in the ABg Scatter Data Catalogs; see "ABg Scatter Data Catalogs" for more information. The TIS is not generally 1.0; the actual value is listed in the output plot. The ABg data may be scaled with wavelength; if a reference wavelength is defined in the catalog. ABg data is also generally a function of incident angle.

Zemax interpolates the BSDF data between the two points with angles of incidence closest to that of the defined incident angle.

ABg File: All ABg profiles specified in the ABGF file must be defined in the currently loaded ABg Data File (see "ABg Data File" in the user's manual). Each ABg profile is treated as it would be under the ABg scatter model. Thus, ABg data for each profile will be scaled with wavelength if a reference wavelength is defined in the ABg Scatter Data Catalog for that profile (see "ABg Scatter Data Catalogs"), and the ABg data for each profile will be interpolated between the two points with angles of incidence closest to that of the defined incident angle. The wavelength scaling and interpolation over angle of incidence will be performed for each profile specified in the ABGF file before all profiles are summed. The TIS for the sum of the ABg profiles (whether it be the weighted or absolute sum) is not generally 1.0; the actual value is listed in the output plot.

BSDF: The BSDF values are defined in an input file; see "BSDF scattering" in the user's manual. The TIS is not generally 1.0; the actual value is listed in the output plot. As described in detail in the Knowledge Base, Zemax performs limited interpolation of the input data. Thus, the input data should have sufficient resolution (specifically

with regards to the incident, azimuthal, and radial angles) to accurately describe any variations of importance in the BSDF data. For BSDF files with low angular resolution, the resolution of the output plot will generally be limited by the data file itself rather than the number of radial and azimuthal bins used in the plot.

IS Scatter Catalog: The BSDF values are defined in an input file generated by the Radiant Zemax IS-SA device; see "IS Scatter Catalog scattering" in the user's manual. This model is functionally very similar to the BSDF scattering model described above, but with data contained in a high resolution binary file rather than an ASCII text file. The spectral variation of the scattering distribution may also be present in an ISX file. As with the BSDF scattering model, the TIS is not generally 1.0; the actual value is listed in the output plot.

User Defined: Zemax cannot determine the exact plot of the BSDF for user defined scattering. As an alternative, a number of identical specular rays (corresponding to the input value for the incident angle) are scattered by the DLL, and the resulting scatter angles are then binned and plotted. BSDF plots made in this way are typically noisy in a complex way that depends upon the nature of the scattering function defined by the DLL.

Sources Group

These tools provide access to different source models.
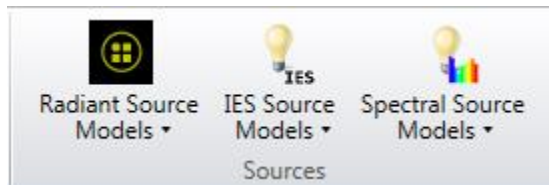
Radiant Source Models
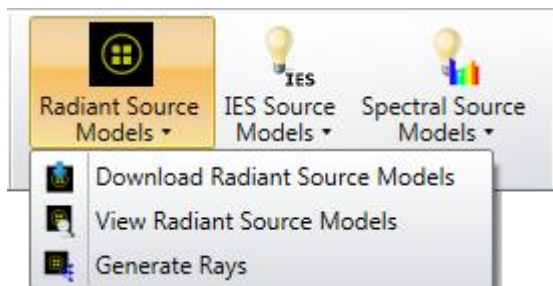
Radiant source models are the most accurate method of modeling the near-field and far-field output of real light sources in optical and illumination design software. Containing angular, spatial, and wavelength data, RSMs are detailed measurements of virtually any type of source, including LEDs, fluorescent lights, arc, tungsten halogen, and HID lamps.

Download Radiant Source Models

This feature is only available in the Premium edition of Zemax. Data download is only available if support on your Zemax key is current.

Browses the online Radiant Source Model library, and downloads Radiant Source Model files.

Manufacturer   Select either all or a specific lamp manufacturer.

Lamp Type   Select either all or a specific lamp type.

File   Selects the Radiant Source Model lamp file. The name presented is a brief description of the source, the actual file name is displayed below this control.

Discussion

This feature provides access to the online library of Radiant Source Models. Many source models from various manufacturers are available for download and use in Zemax-Premium. Once these source models are downloaded, Zemax can generate ray sets based upon the measured lamp data. For information on generating ray sets see "Generate Radiant Source Model Rays" .

The catalog of available source models is maintained on the local machine. To update this catalog, press the "Download Source Catalog" button. This will download the latest catalog data from the online library, and then the available source models will be shown in the dialog box controls.

Descriptive lamp parameters, such as Package Size, Viewing Angle, Test Current, and Minimum Flux are taken from manufacturer's published data. It is up to the user to independently verify stated lamp parameters. Minimum Flux is measured at the stated Test Current.

Once a source model is selected, press "Download Source Model" to begin the download. Note that Radiant Source Model files are large files, and may take a long time to download, depending upon the source model file size and the connection speed.

To close this dialog box, press Close.

To close the dialog box and move directly to the ray generation dialog, press Generate Rays.

Radiant Source Model™ Viewer

This feature is only available in the Premium edition of Zemax.

Displays source images contained within Radiant Source Model™ files. These files may be downloaded from the online data base of source models, see "Download Radiant Source Models" .
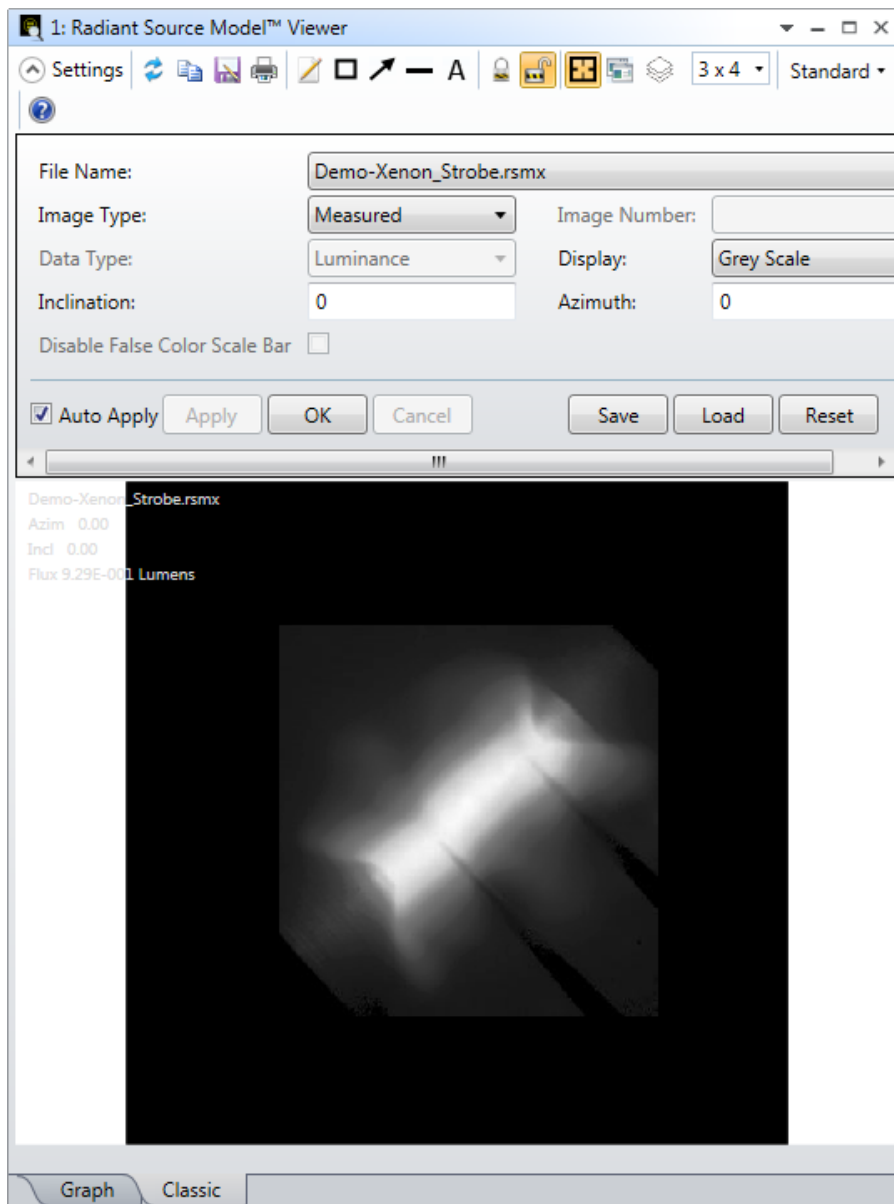
File Name The name of the RSMX file to display the data for.

Image Type Choose Measured (source is lit) or Alignment (source is unlit) images, or a CRI Radar plot (2D Color Rendering Index map for the source, which will only be available for sources that contain full spectral measurements).

Image Number The image number to display for Alignment Images, or the emission direction for which CRI data are calculated..

Data Type Choose True Color or Luminance. True Color is only available for sources which contain color data.

Display If "Luminance" is selected for the Data Type, then the data may be displayed in Grey Scale or False Color. The false color palette used by this feature differs from that normally used by Zemax, but matches that used by the Radiant Zemax ProSource© software.

Inclination The polar angle in degrees measured from the +Z axis. This angle should be between 0 and 180 degrees.

Azimuth The angle in degrees measured around the +Z axis. This angle should be between 0 and 360.

Disable False Color Scale Bar   Only available if the Display is set to False Color. If checked, then the false color bar indicating the color levels is not shown.

Discussion

This feature is used to display actual source images, alignment images, and CRI radar maps for the specified Radiant Source Model™.

When displaying Measured images (i.e. images of a lit source), the image is shown for the specified Inclination and Azimuth angles. These angles define points on an imaginary sphere. The Inclination is the polar angle between the +Z axis and the view direction. The Azimuth is defined as the clockwise angle from the +X direction of the projection of the view direction on the XY plane. The displayed image is the closest measured image to the specified view angle.

Once a measured image has been displayed, other measured images may be accessed using the arrow keys, without having to subsequently open up the settings dialog for the feature. If the arrow keys were used to change the measured image, the values for the Inclination and Azimuth angles corresponding to the measured image that is currently being viewed will appear in the settings dialog whenever it is subsequently opened.

Alignment images will only be available at a select number of Inclination and Azimuth angles. Thus, for alignment images the options to specify Inclination and Azimuth angles are greyed out. However, the angles corresponding to each alignment image may be viewed within the settings dialog for the feature prior to displaying the image.

Color Rendering Index (CRI) data are only available for sources that contain full spectral measurements. For such sources, a 2D polar plot of the CRI data may be viewed. This data is calculated for both the Forward and Rear emission directions of the source, and the user may choose which direction to view (under the "Image Number" input when "Image Type" is set to CRI Radar).

The RSMX files must be stored in the <data>\Objects\Sources\Radiant Source Models (see "Folders" in the user's manual).

Generate Radiant Source Model Rays

 This feature is only available in the Premium edition of Zemax.

This feature generates ray sets in the form of Source Files from Radiant Source Model files. To download

Radiant Source Model files, see "Download Radiant Source Models"

Radiant Source File   The name of the Radiant Source Model RSMX file to use in the ray generation.

Spectrum Type   If the RSMX file contains measured spectrum data, then the only available option is "Measured" (i.e. to use the measured data in defining the ray wavelengths). Otherwise, the spectrum must be defined. See the Discussion for details.

Spectrum File   If the Spectrum Type is Spectrum file, then the name of the spectrum file may be selected. This file may also be found using the Browse button.

W a v e l e n g t h , Spectral Width The center wavelength and spectral width in micrometers for the narrowband Spectrum Type.

Temperature (K) The temperature in Kelvins for the Blackbody Spectrum Type.

Minimum/Maximum Wavelength   The minimum and maximum wavelength to generate (in micrometers). These controls define the limit of the spectral bandwidth.

Minimum/Maximum Azimuth Angle   The azimuthal angular range (in degrees) over which rays will be generated. The values must be between 0 and 360.

Minimum/Maximum Inclination Angle   The inclination angular range (in degrees) over which rays will be generated. The values must be between 0 and 180.

Ray Origination   The shape of the line, surface, or volume to use for the starting location for all generated rays. This should be chosen to mimic the shape of the source itself. If all optical components lie in the far-field of the source, the method chosen is not important. The raytrace results are identical if all optical components lie outside the surface or volume the rays originate from. Normal plane is the default selection. See the Discussion for details.

Radius   This setting is used only when the selected Ray Origination method is "On Sphere", "On X Cylinder", "On Y Cylinder", or "On Z Cylinder". The parameter defines the radius (in millimeters) of the sphere or cylinder from which rays will originate.

End Caps?   This option is available only when the selected Ray Origination method is "On X Cylinder", "On Y Cylinder", or "On Z Cylinder". If this option is selected, then the positions of the cylindrical end caps (along the cylinder axis) may be specified via the "Lower Limit" and "Upper Limit" inputs, and any rays that originate beyond these end cap limits will be moved back along their vector trajectories to the corresponding limit plane.

Lower/Upper Limit   This option is only available when the selected Ray Origination method is "On X Cylinder", "On Y Cylinder", or "On Z Cylinder", and the "End Caps?" option has been selected. The input values represent the positions for the lower and upper end caps of cylinder (along the cylinder axis) in millimeters.

Minimum/Maximum X/Y/Z   This option is only available when the selected Ray Origination method is "On Box". Values represent the X, Y, and Z positions (in millimeters) of the corners of the box from which rays will originate.

Source Translation X/Y/Z These values are used to offset (translate) the rays along the particular axes (X, Y, and Z) after the rays are both generated via the selected Ray Origination method and optionally rotated via the "Source Rotation" inputs. Input values are in millimeters.

Source Rotation X/ Y/Z   These values are used to rotate the generated rays about the particular axis (X, Y, and Z) after the rays have been generated via the selected Ray Origination method. The rotations are applied in the following order: rotate about X, then about Y, then about Z. The input values are in degrees.

Number Of Rays The number of rays to generate. The number of rays actually generated from the RSMX file will not exactly match this input value (though it will be very close) as a result of the algorithm used to generate rays from the RSMX file.

Output Source File   The name of the output file to use. The output file is always stored in the

<data>\Objects\Sources\Source Files folder (see "Folders" in the user's manual). This feature only supports the SDF file format and the extension SDF should always be used.

Discussion

The Spectrum Type is only used if the Radiant Source Model file is flux data only, with no associated spectral data. The available Spectrum Types are:

- Spectrum file: An SPCD file that defines the relative wavelengths and weights for the spectrum. For details on the SPCD file format see "Defining a spectrum file" in the Sources section of the Object Properties.

- Blackbody: A spectrum based upon the blackbody curve. The temperature in Kelvins determines the spectrum.

- Narrowband: A narrow Gaussian spectral band based upon a center wavelength and Spectral Width, which is twice the standard deviation of the distribution.

The minimum/maximum inclination and azimuth angles allow selection of rays within an angular subset of the measured data. The inclination angle is measured from the +Z axis. The valid range of the inclination angle is from 0 to 180 degrees. The azimuth angle is measured counter-clockwise from the +X axis in the XY plane, as shown in the diagram below. The valid range of the azimuth angle is 0 to 360 degrees.
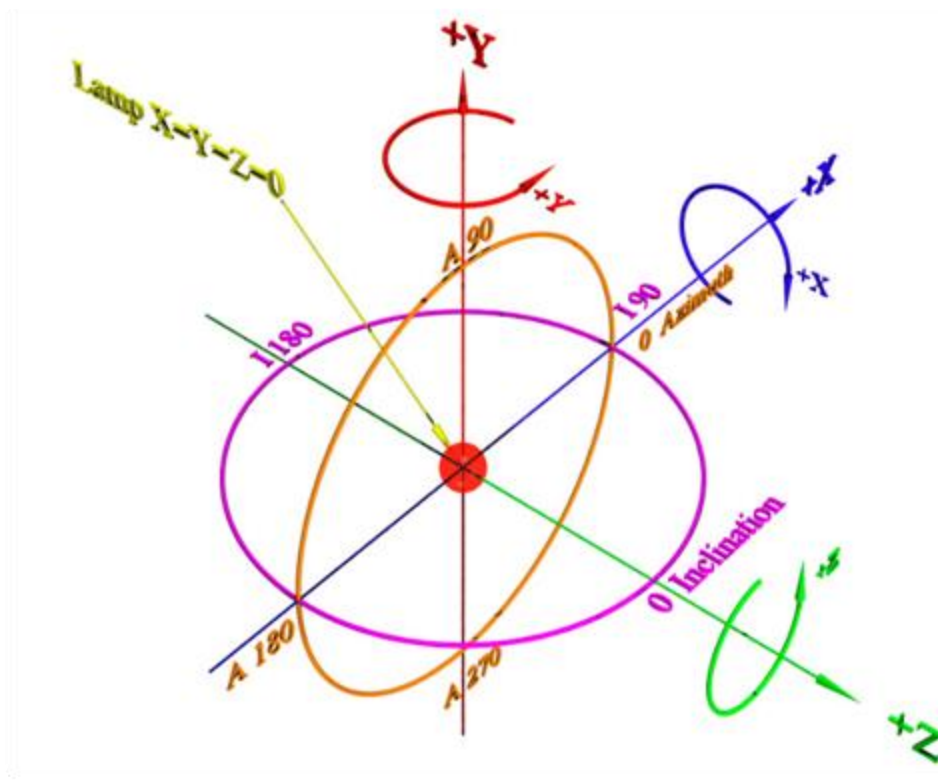
The various Ray Origination methods are described below:

- Normal Plane: Each ray originates from a plane that intersects the origin of the source and is normal to the ray direction. This method generates ray sets with origination points furthest from the active area of the source. Do not use this method if optical surfaces in your design are positioned closer than the surface of revolution of the light source.

- Closest to X/Y/Z Axis: Each ray originates from the point where it intersects or is closest to an axis (X/Y/Z). This is a good method to use for optical designs which have a parabolic reflector along the corresponding axis. Do not use this method with highly collimated light sources that produce rays parallel to the corresponding axis.

- On Sphere: Each ray originates from the point where it intersects a sphere that is centered on the origin of the source and has a radius given by the "Radius" input. If a ray does not intersect this sphere, the point closest to the origin is selected for the ray. This method can be used to locate ray origination points between the lamp structure (e.g. lamp filament) and secondary optics for designs in which the lamp structure is well- encapsulated by a sphere. As such, this method is useful for optical designs in which interactions between rays and the lamp structure may be important (thus making it necessary to originate all rays on a surface outside of the lamp structure).

- On X/Y/Z Cylinder: Each ray originates from the point where it intersects with a cylinder of infinite length (unless the "End Caps?" option is selected, in which case the cylinder length is set by the "Lower" and "Upper Limit" inputs) and a radius given by the "Radius" input, with the direction of the cylinder axis along the X/Y/Z axis. If a ray does not intersect this cylinder, the point closest to the cylinder axis is selected for the ray. This is a good method to use with elongated sources (e.g. filaments, fluorescent tubes). This method can be used to locate ray origination points between the lamp structure and secondary optics for designs in which the lamp structure is well-encapsulated by a cylinder, and is therefore useful for optical designs in which interactions between rays and the lamp structure may be important.

- On X/Y/Z Plane: Each ray originates from the point where it intersects the designated plane (X/Y/Z). This method should only be used for directional sources in which light is emitted roughly normal to the designated plane. For such sources, this method is good for designs in which secondary optics are placed both along the axis normal to the designated plane and close to that axis.

- On Box: Each ray originates from the point where it intersects a box whose dimensions are given by the "Minimum" and "Maximum X/Y/Z" inputs. If a ray does not intersect this box, the point closest to the center of the box is selected for the ray. This method can be used to locate ray origination points between the lamp structure and secondary optics for designs in which the lamp structure is well-encapsulated by box, and is therefore useful for optical designs in which interactions between rays and the lamp structure may be important.

Once the settings are selected, press Generate Rays to being the ray generation. To use the generated ray set, add a Source File to the NSC Editor and select the ray file name from the Object Properties dialog. For more information, see "Source File".

The RSMX files must be placed in the <data>\Objects\Sources\Radiant Source Model Files folder (see "Folders" in the user's manual).

Note that this feature may pause occasionally when large numbers of rays are being generated. This is because Zemax can generate rays faster than they can be written to disk, and once the disk cache is full ray generation must pause until the disk can be written to again. The use of Solid State Disks is highly recommended if you intend to generate large ray sets often.

IES Source Models
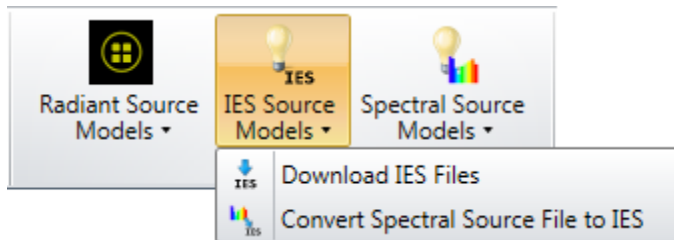
IES Source Models contain angular data only and are suitable if the source is in the far field of the optical system.

Download IES File

This feature is only available in the Premium edition of Zemax. Data download is only available if support on your Zemax key is current.

This feature downloads IES source model files from the Zemax online source model library.
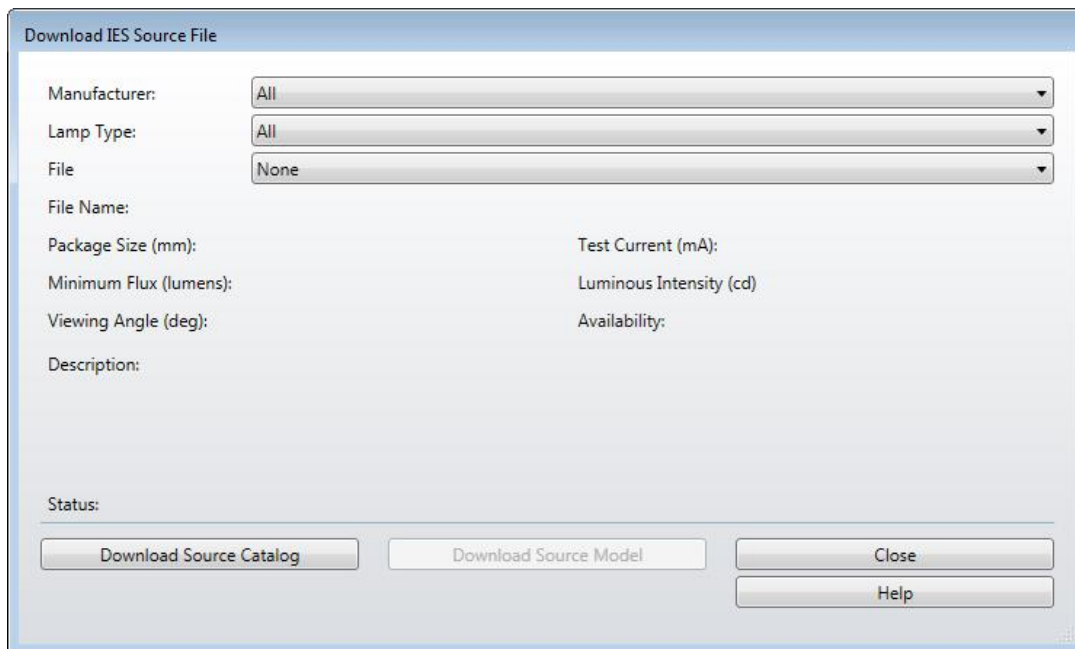
Discussion

This feature is nearly identical to the Download Radiant Source Models feature. See "Download Radiant Source Models" for details.

Convert SDF to IES

This feature is only available in the Premium edition of Zemax.

To convert a spectral color format (SDF) file into an IES file.
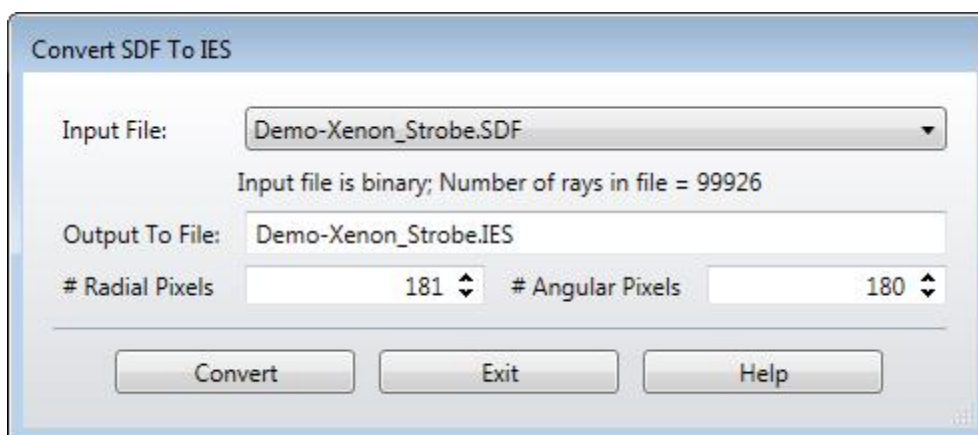
Input File   The name of the SDF file to be converted.

Output File   The name of the file containing the converted results. The file extension must be .IES.

 # Radial Pixels   The number of radial data points to use in the resulting IES file (corresponds to vertical angles in the IES nomenclature).

 # Angular Pixels The number of angular data points to use in the resulting IES file (corresponds to horizontal angles in the IES nomenclature).

Discussion

This feature allows a spectral color format file (see "Source File") to be converted to an IESNA file (see "Source IESNA File"). This conversion allows for a faster use of the source data for modeling systems in which the source is in the far-field relative to all optics.

SDF files contain information regarding the spectral distribution of the source, whereas IESNA files do not. The spectral content present in the SDF file is retained during conversion by the creation of a separate spectrum file with an SPCD extension (see "Defining a spectrum file" in the [Sources](#) section of the Object Properties.   ).

To create the IESNA and SPCD files from an SDF file, first select the SDF file. By default the name of the IESNA file will be the same as that of the selected SDF file, simply with a different extension (IES). However, the user may modify the name of the IESNA file if desired. The name of the spectrum file will always be the same as that of the IESNA file, simply with a different extension (SPCD).

Next, the user needs to specify the resolution to use in the IESNA file, i.e. the number of vertical and horizontal angles (Zemax refers to these as Radial and Angular angles, respectively). The user should be careful not to specify unreasonably large values, especially when converting from sparse SDF files. For example, the default pixelation values specified in the tool (181 vertical angles, 180 horizontal angles) are reasonable when converting SDF files containing around 100,000 rays or more. For smaller ray sets, courser angular resolution is recommended.

Once the input file has been selected and the output file name and pixelation have been specified, the IESNA and spectrum files are created by selecting the "Convert" button. The IESNA file will be placed in the <data>\Objects\Sources\IESNA\ folder, while the spectrum file will be placed in the <data>\Objects\Sources\Spectrum Files\ folder.

To close this dialog box, press "Exit".

Spectral Source Models
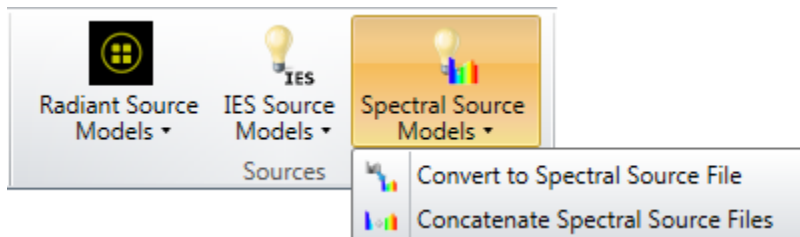
Tools for adding spectral data to sources that do not contain it.

Convert to Spectral Source File

This feature is only available in the Premium edition of Zemax.

To convert a source file that does not contain wavelength information (i.e. a source file with a .DAT extension) to one that contains such information.
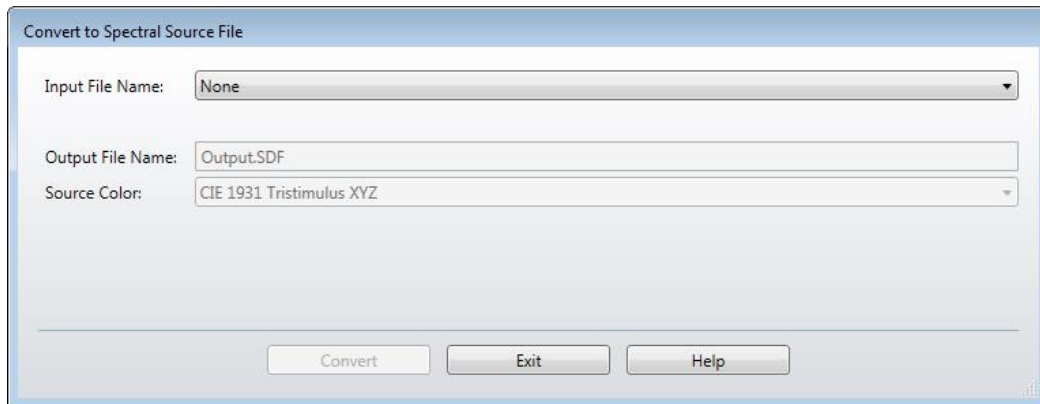
Input File Name The name of the DAT file to be converted.

Output File Name The name of the file containing the converted data. The file extension must be .SDF

Source Color The model used to define the spectral distribution of the source. The ten options provided for the source color model are described in "Defining the color and spectral content of sources" .

X, Y, Z; x, y; R, G, B; Temp (K); u'v' Six of the models require additional numeric inputs for characterizing the source color. The appropriate inputs should be entered based on the selected model (e.g. X, Y and Z values for the CIE 1931 Tristimulus XYZ model). No additional numeric inputs are required for the System Wavelengths, Uniform Power Spectrum, D65 White, or Spectrum File models. For any model requiring additional numeric inputs - other than Black Body Spectrum - Zemax must perform fitting to determine the corresponding spectrum (this fitting is also required for the D65 White model). More details on the fitting algorithm are provided in "The spectrum fitting algorithm" .

Spectrum The number of wavelengths to use in characterizing the spectral distribution of the source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

Wavelengths From The minimum wavelength (in microns) to use in characterizing the spectral distribution of the source.

The user has no control over this option when selecting the Spectrum File model (though the appropriate value from the spectral file will be read into the dialog). When select the System Wavelength model, the title of this setting will change to "Wavenumber", with the input value corresponding to the wavenumber to use for defining the spectral distribution (zero means polychromatic, i.e. all wavelengths defined in the Wavelength Data dialog box will be used according to their relative weights).

To The maximum wavelength (in microns) to use in characterizing the spectral distribution of the source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

Spectrum File The name of the file containing the spectral data for the source. This option is only available when selecting the Spectrum File model. More information on the file format is provided in "Defining a spectrum file" in the [Sources](#) section of the Object Properties.

Discussion

This feature allows spectral information to be added to a source file, converting the file from a flux only format file to a spectral color format file (see "Source File"). The spectral data for the source are added via specification of a source color model and its corresponding parameters. Zemax assigns a wavelength to each ray in the input file based on a Monte Carlo sampling algorithm, such that the overall color of the source matches the input specification. The original data, with wavelengths now included, are then written to a new file (with the appropriate .SDF extension), while the initial input file (.DAT extension) is retained.

To create an SDF file from a DAT file, first select the desired DAT file. By default the name of the SDF file will be the same as that of the selected DAT file, simply with a different extension. However, the user may modify the name of the SDF file if desired; thus, a single DAT file could be used to create different SDF files with different spectral distributions.

Once the input DAT file has been selected (and the corresponding SDF file name has been provided), a source color model is specified. The remaining inputs are then provided based on the selection of the source color model.

Once the file and color model have been selected, the SDF file is created by selecting the "Convert" button. The resulting file will be placed in the <data>\Objects\Sources\Source Files\ folder.

To close this dialog box, press "Exit".

Concatenate Spectral Source File

This feature is only available in the Premium edition of Zemax.

To concatenate either two or three spectral color format (SDF) files into a single SDF file.
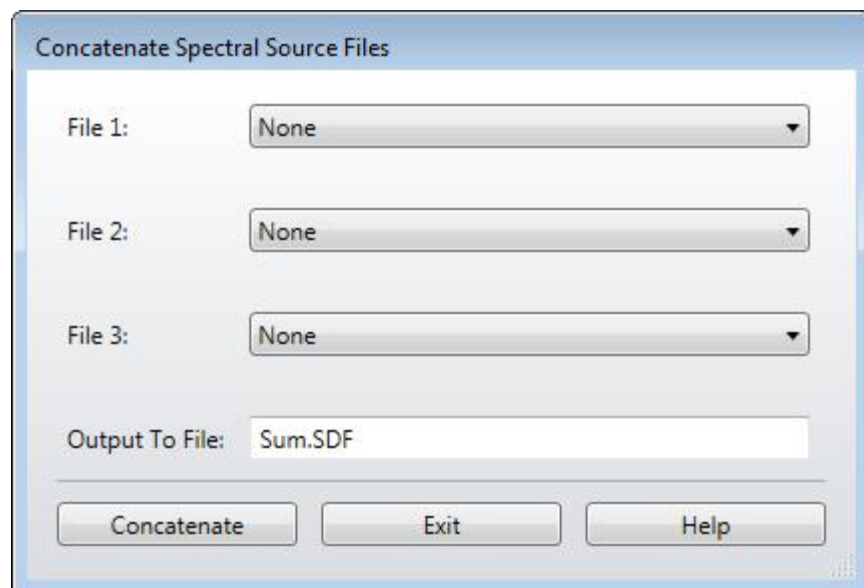
File 1   The name of the first SDF file to be used in the concatenation. None may be selected, if the user only wishes to concatenate two files (see details below).

File 2   The name of the second SDF file to be used in the concatenation. None may be selected, if the user only wishes to concatenate two files (see details below).

File 3   The name of the third SDF file to be used in concatenation. None may be selected, if the user only wishes to concatenate two files (see details below).

Output File   The name of the output file containing the concatenated results. The file extension must be .SDF.

Discussion

This feature allows either two or three spectral color format files (see "Source File") to be concatenated into a single file. The concatenation will only be allowed if all of the input files are binary format. In addition, all of the files must use the same dimension units, and the translation, rotation, and scale factors in each file must be identical in order for the concatenation to proceed. This feature is best-suited for situations in which different color filters were used to make measurements for the same lamp and measurement geometry and the user now wishes to combine the filtered data (each present in a separate source file) into a single file.

To create the concatenated file, first select the three desired input files ("File 1", "File 2", and "File 3"). To concatenate just two files, select "None" for one of the file names. Note that if "None" is selected for more than one of the file names, an error will be issued and the concatenation will not proceed. Although a default name for the concatenated file is provided ("Sum.SDF"), a user defined name should be provided for the output file.

Once the input files have been selected and the output file name has been specified, the concatenated file is created by selecting the "Concatenate" button. The output file will be placed in the <data>\Objects\Sources\Source Files\ folder.

To close this dialog box, press "Exit".

Source Viewers Group

These features are used to display data for source models.
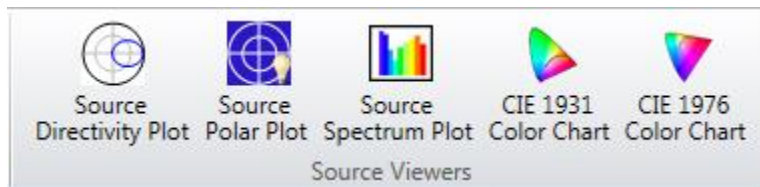
Directivity Plot

This feature is only available in the Premium edition of Zemax.

Displays a directivity diagram for IES, LDT, and RSMX source files.
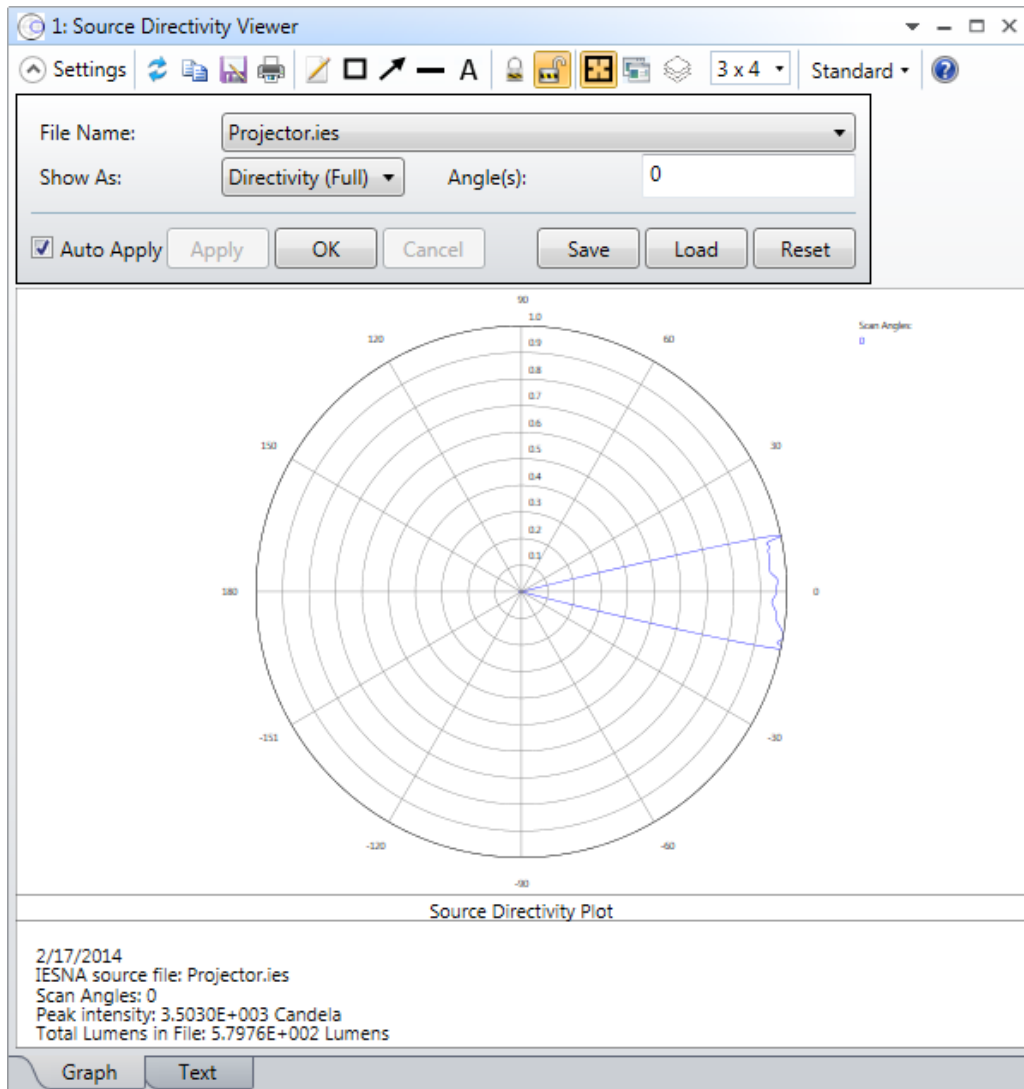
File Name The name of the IESNA IES, EULUMDAT LDT, Radiant Source Model RSMX, or rayset SDF file containing the data to be displayed.

Show As The directivity data may be displayed either in a Half or Full Sphere.

Angle(s) This control supports multiple angles separated by commas. For example, the Angle(s) list may be 0, 45, 90, 135. All of the data for all angles are then shown on the same plot. A maximum of 5 angles may be listed.

Discussion

The directivity cross-sections lie in a plane that contains the local Z axis and a point rotated counter-clockwise around Z from the +X direction (when the point is aligned with the +X axis the angle is zero).

IES files must be placed in the <data>\Objects\Sources\IESNA folder.

LDT files must be placed in the <data>\Objects\Sources\EULUMDAT folder.

RSMX files must be placed in the <data>\Objects\Sources\Radiant Source Model Files folder. SDF files must be placed in the <data>\Objects\Sources\Source Files folder.

For more information on folders, see "Folders" in the user's manual.

Polar Plot

This feature is only available in the Premium edition of Zemax.

Displays a hemi- or full-sphere map of the source luminance for IES, LDT, and RSMX source files.
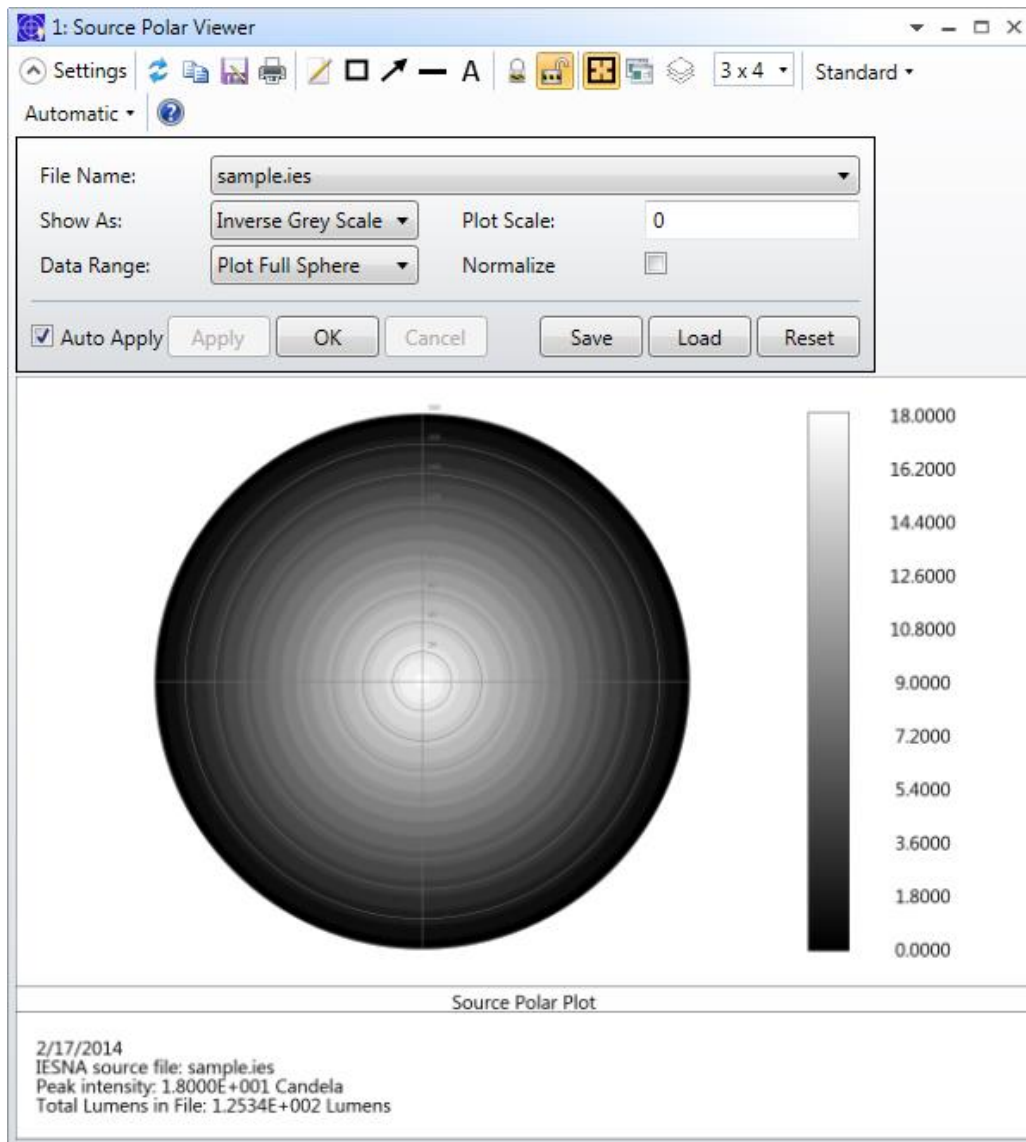
File Name   The name of the IESNA IES, EULUMDAT LDT, Radiant Source Model RSMX, or rayset SDF file containing the data to be displayed.

Show As   The data may be displayed as either Grey Scale, Inverse Grey Scale, False Color, or Inverse False Color.

Plot Scale   This control sets the maximum upper limit of the plot scale.

Data Range   The polar data may be displayed in a Full Sphere (out to a polar angle of 180 degrees) or in a Hemisphere (out to a polar angle of 90 degrees).

Normalize   Normalizes all data in the file to a peak value of 1.0 Candela.

Discussion

IES files must be placed in the <data>\Objects\Sources\IESNA folder.

LDT files must be placed in the <data>\Objects\Sources\EULUMDAT folder.

RSMX files must be placed in the <data>\Objects\Sources\Radiant Source Model Files folder. SDF files must be placed in the <data>\Objects\Sources\Source Files folder.

For more information on folders, see "Folders" in the user's manual.

Spectrum Plot

This feature is only available in the Premium edition of Zemax.

Displays the relative intensity vs. wavelength for a defined spectral distribution.

Source Color The model used to define the spectral distribution of the source. The ten options provided for the source color model are described in "Defining the color and spectral content of sources".

X, Y, Z; x, y; R, G, B; Temp (K); u'v' Six of the models require additional numeric inputs for characterizing the source color. The appropriate inputs should be entered based on the selected model (e.g. X, Y and Z values for the CIE 1931 Tristimulus XYZ model). No additional numeric inputs are required for the System Wavelengths, Uniform Power Spectrum, D65 White, or Spectrum File models. For any model requiring additional numeric inputs - other than Black Body Spectrum - Zemax must perform fitting to determine the corresponding spectrum (this fitting is also required for the D65 White model). More details on the fitting algorithm are provided in "The spectrum fitting algorithm" .

Spectrum The number of wavelengths to use in characterizing the spectral distribution of the source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

Wavelengths From   The minimum wavelength (in microns) to use in characterizing the spectral distribution of the source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

To   The maximum wavelength (in microns) to use in characterizing the spectral distribution of the source. The user has no control over this option when selecting the System Wavelengths or the Spectrum File models (though in the latter case the appropriate value from the spectrum file will be read into the dialog).

Spectrum File   The name of the file containing the spectral data for the source. This option is only available when selecting the Spectrum File model. More information on the file format is provided in "Defining a spectrum file" in the Sources section of the Object Properties. .

Bar Chart   If selected, the data will be displayed as a bar chart.

Discussion

This feature is used to visualize the spectral distribution of a source defined in non-sequential mode. The distribution may be defined according to any one of ten models, each of which are described in detail in "Defining the color and spectral content of sources" . The results are shown as either check marks or bars at each of the discrete wavelengths for the distribution. The check marks or bars are colored according to the closest RGB value for the corresponding wavelength (if no RGB value exists for a particular wavelength a black color is used for that data point). More details on how the wavelengths and weights are determined for any given source color model may be found in "The spectrum fitting algorithm" . The input distribution and the Tristimulus values corresponding to the fit for that distribution (as determined by the spectrum fitting algorithm) are displayed in the text section of the analysis. At the bottom of this section are the Color Rendering Index (CRI) and Color Correlated Temperature (CCT) for the distribution.

CIE 1931 Color Chart

 This feature is only available in the Premium edition of Zemax.

This feature displays color chart based upon the CIE 1931 color spaces.
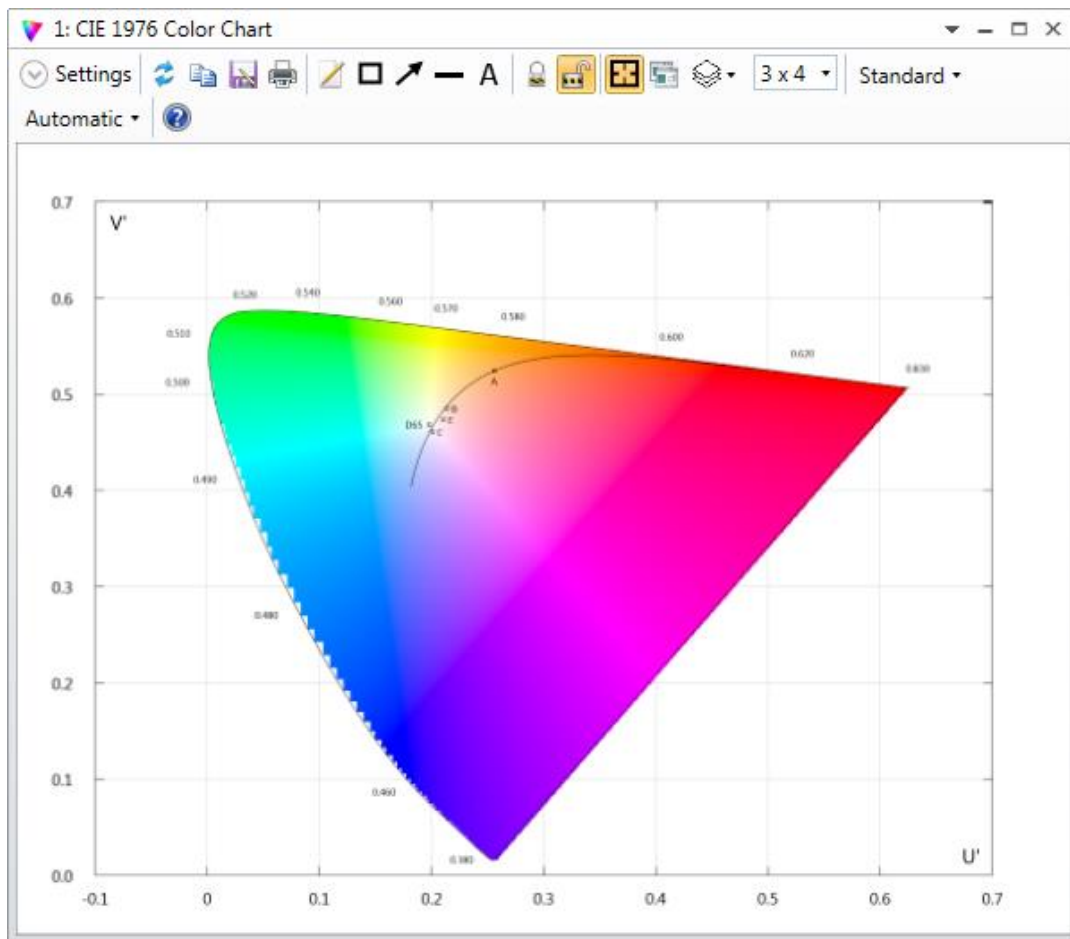
Discussion This chart is provided for reference only, and is not based upon any properties of the current system. See also "Defining the color and spectral content of sources" in the manual.

CIE 1976 Color Chart

This feature is only available in the Premium edition of Zemax.

This feature displays color chart based upon the CIE 1976 color spaces.

Discussion This chart is provided for reference only, and is not based upon any properties of the current system. See also "Defining the color and spectral content of sources" in the manual.

Part Designer Tab
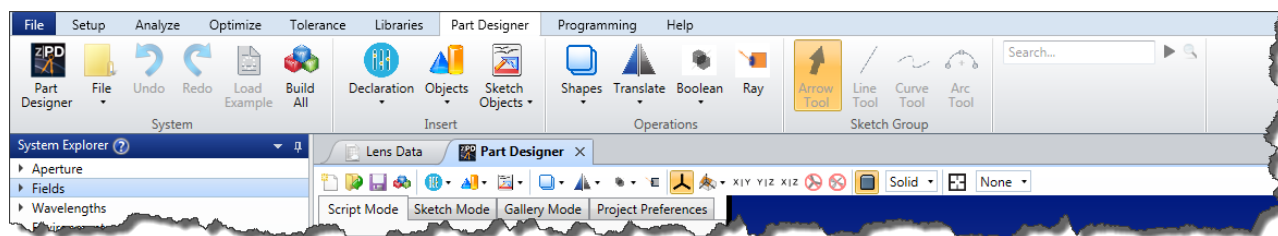
The Zemax Part Designer (ZPD) is an interface that allows users to create and manipulate geometries in an environment separate from Zemax. Special features such as the "Ghost View" and On-The-Go Geometry creation allow users to quickly build, assess, and modify individual steps leading up to the final object. Using the ZPD console, users may quickly modify and add commands and operations to a geometry model and generate results within moments. Overall, the Part Designer allows for a streamlined process to incorporate solid 3-D geometry into the user's Zemax non-sequential designs.
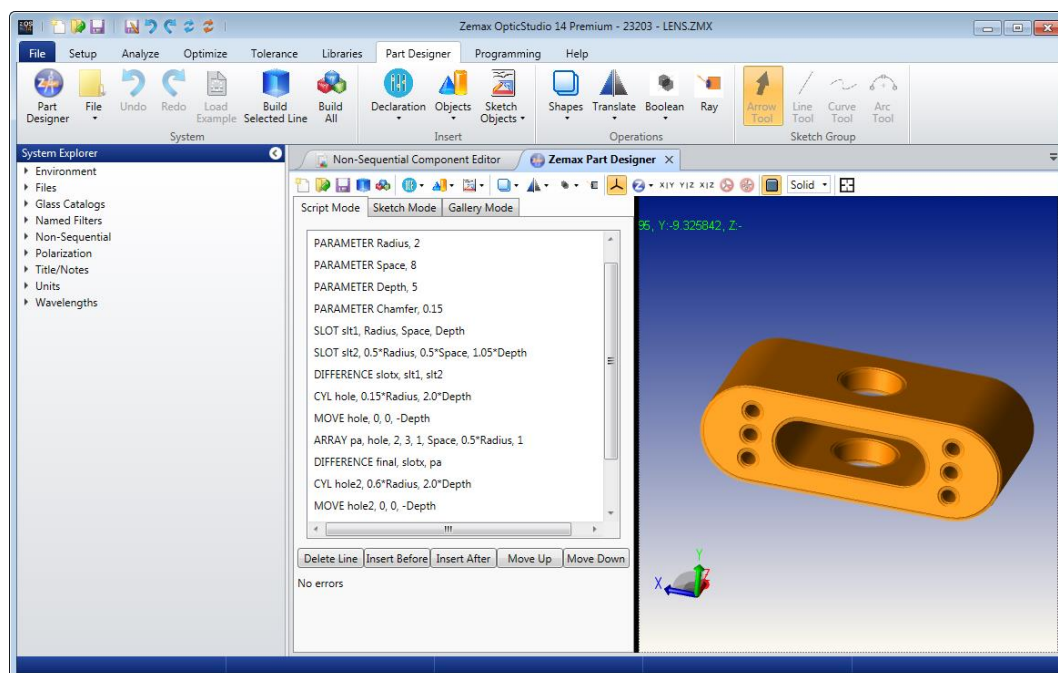
Script, Sketch, and Gallery Modes and Project Preferences

The Zemax Part Designer has three main modes of operation: the Script, Sketch, and Gallery Modes. This is also where the Project Preferences for the Part Designer are set.
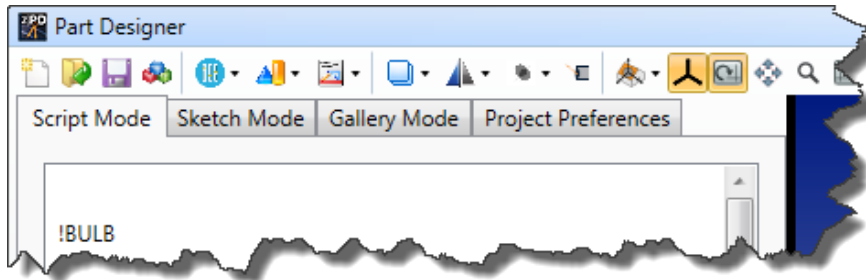
Script Mode

The Script Mode is the foundation of the Part Object. This mode acts as a standard text editor and allows users to write a script to generate the user's desired geometries.
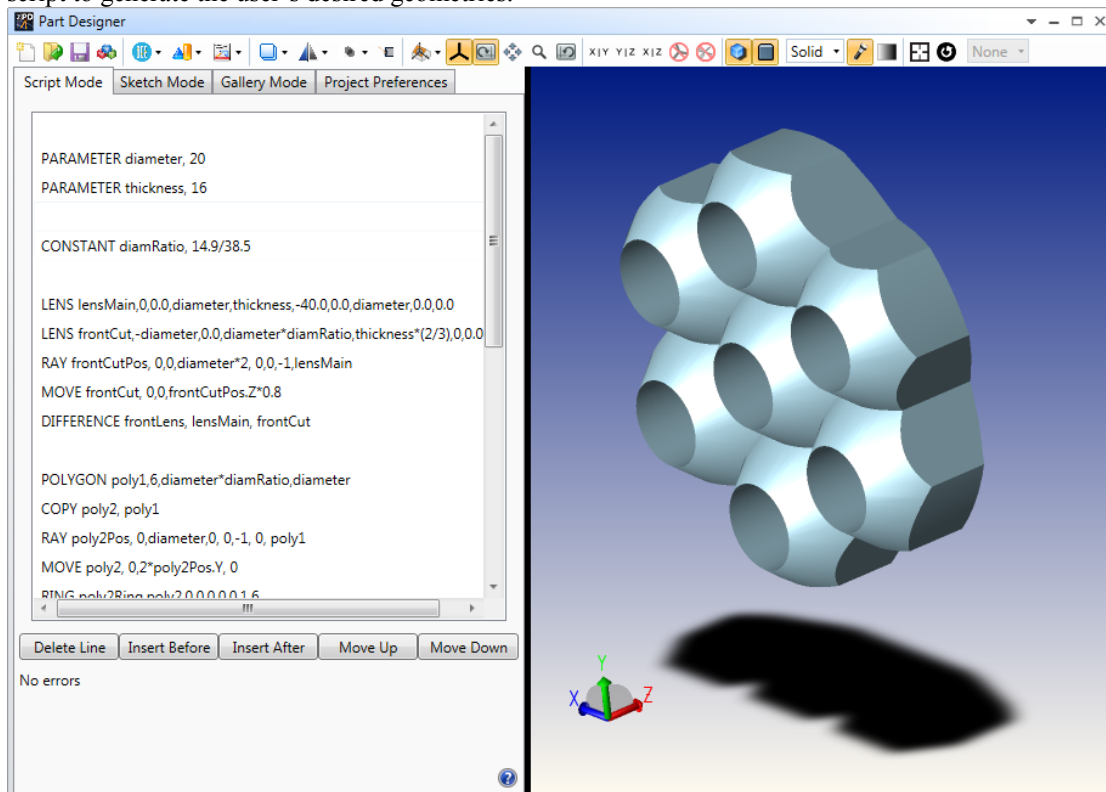
Sketch Mode

The Sketch Mode is a simple yet robust toolset that allows users to define sketches. These sketches are used in conjunction with commands listed under "Sketch Object" of the application toolbar.
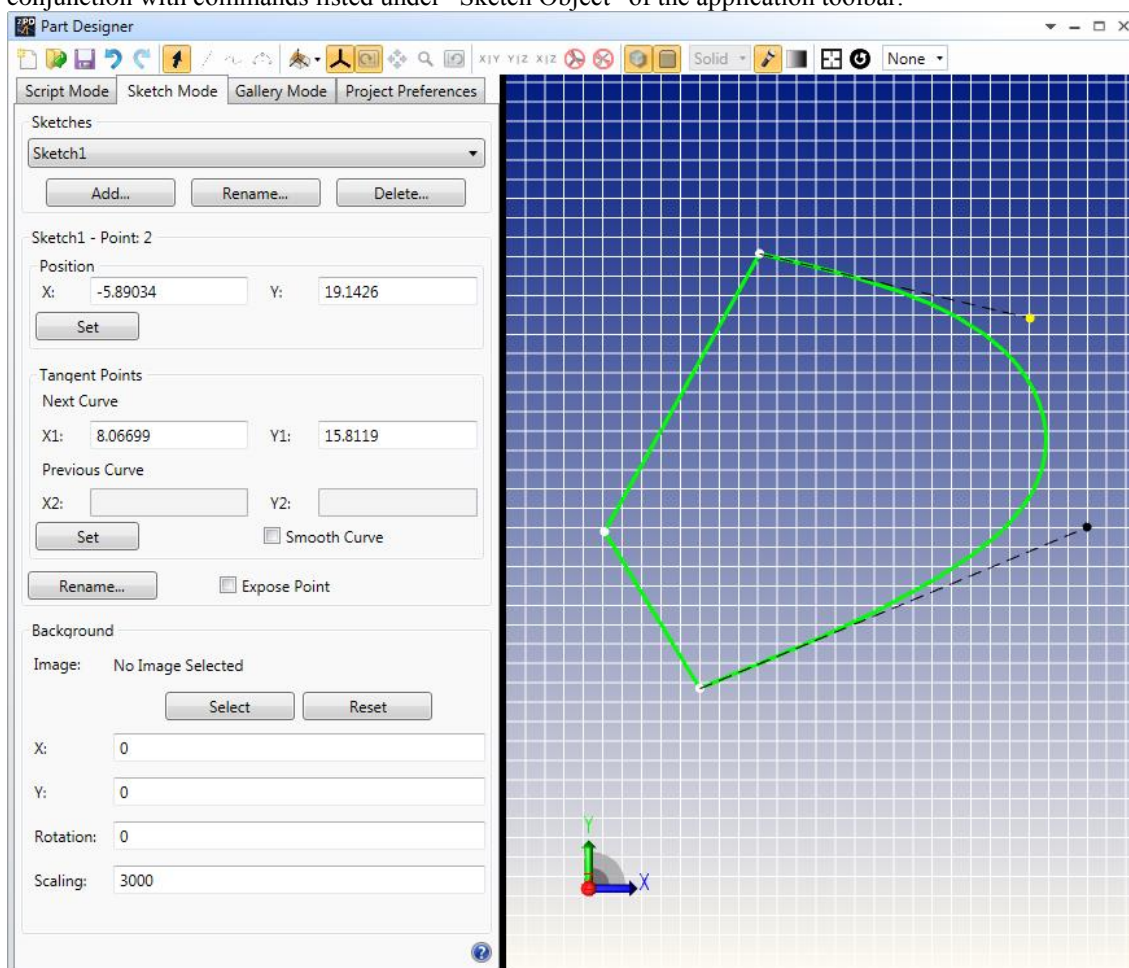
Gallery Mode

The Gallery Mode explores the contents of the default Zemax Part Designer directory. The gallery allows users to sort the files by application (Collectors, Sources, Fixtures, Misc…) or by name. Users may double click each file listed to view a brief description of the part.
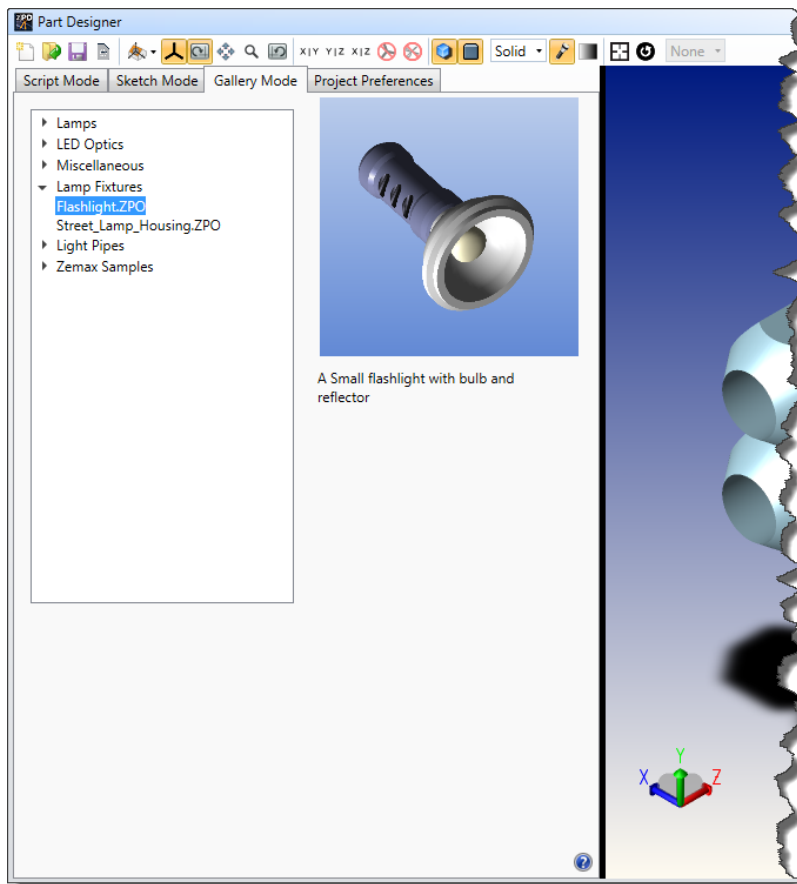
Project Preferences

The Project Preferences tab allows users to modify settings for the current part and the ZPD 3D display.
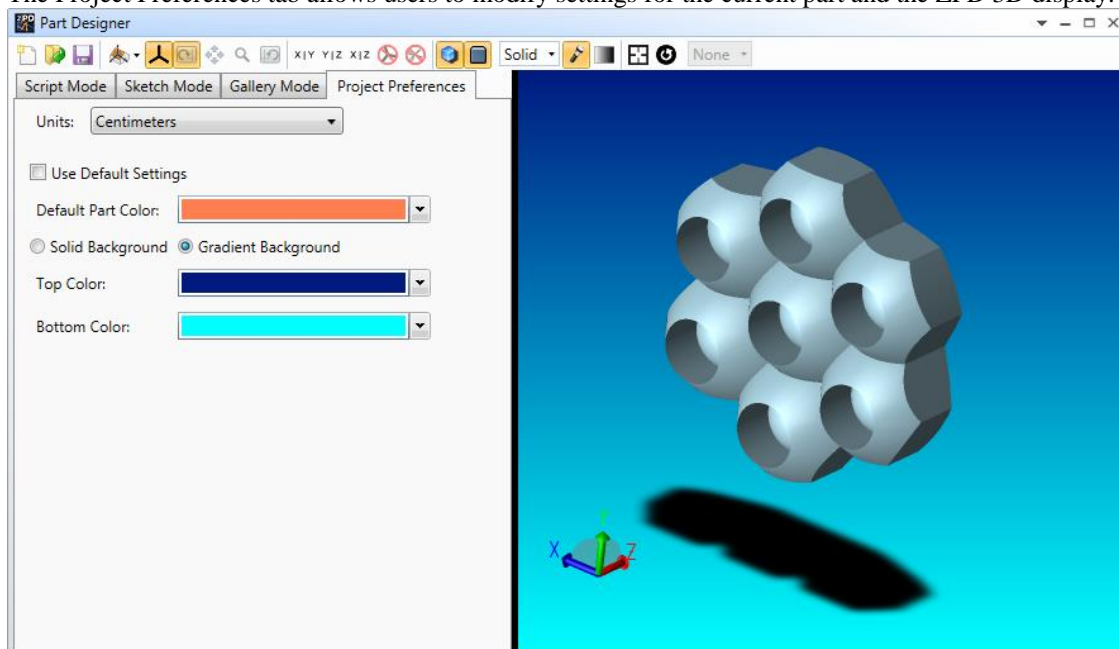
Units

By default, ZPD will utilize millimeters as a unit of measurement. Saving a part will associate a part with the selected unit type. This will not change until the selected unit type is changed and the part is saved again.

When opening a part in ZPD, ZPD will set its unit type to utilize that of the current file. Once the file is closed, ZPD will return the unit settings to what it was set to prior to opening the file.

When opening a part in Zemax, a warning will be issued if the unit type associated with the part file differs from the current Zemax File's system lens unit. If the units differ, Zemax will not perform a conversion.

ZPD parts exported in a CAD format will utilize the selected unit type. If a part is opened and exported within Zemax, Zemax will use the units specified by the system's lens units.

Use Default Settings

If checked, the remaining settings will be locked to their default values and cannot be changed by the user. In order to choose custom settings, uncheck this box.

Default Part Color

This setting controls the color of parts designed in ZPD before any COLOR command is applied. This only controls the color within ZPD itself; a ZPD object in Zemax is not affected by this setting. Note: changes to this setting will not be applied until the script is re-built (i.e. Build All).

Background Color

Solid Background: A single color chooser that allows users to choose from either a predefined palette of colors, or a specific RGB value (under Advanced) to be used as a solid background.

Gradient Background: Two sets of color choosers, each representing the RGB values of vertical gradient background. The top and bottom selections represent the colors of their respective positions of the background gradient.

System Group

Part Designer

Launches Part Designer window

File

Load, Save and export options for Part Designer.
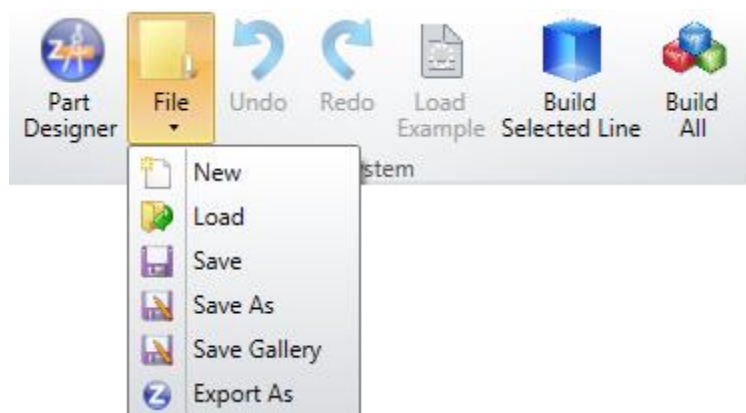
New   Clear the current file and create a new part

Load   Open an existing file

Save   Save the Part Designer file with the current name, overwriting the older version.

Save As Save the design file with a new name, keeping the original version unchanged.

Save Gallery   The "Save Gallery Data" option allows users to save the file in its current state with supplementary information including a user description and bitmap image. This information can be accessed through the Gallery Mode.

Export As   Export parts in a variety of CAD file formats such as .IGS, .STEP, .STL, and .SAT

Undo   Undo last change

Redo   Redo last undo operation

Build Selected Line   Compile selected line

Build All   Compile and display the current model

Insert Group
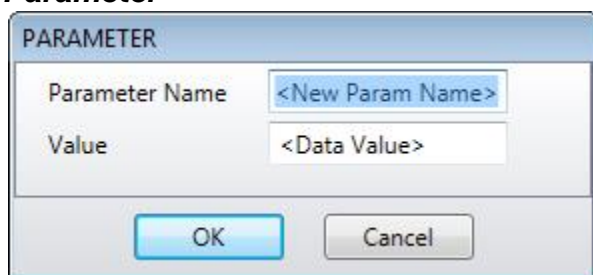
Declaration

Used to store variables for use in Zemax and within the script

The declaration commands are used to define numerical values that may be used elsewhere in the script by reference to the declared value name. There are two types of declarations: parameters and constants. The difference between a parameter and a constant is that a parameter is exposed to the Zemax user interface, and the value defined in the script may be overridden with a different value in the non-sequential component editor.

Parameter

PARAMETER name, value

The PARAMETER command defines the name and numerical value of a constant which can subsequently be used in the place of any numerical value. This named parameter is also automatically exposed to Zemax as a user definable parameter. Changing or optimizing the parameter in Zemax will rebuild the entire object with the new value, and any value defined in Zemax will override the value in the script. The script value will only be used as the "default" value the first time the script is loaded into an NSC object. Parameters may not be defined using values from previously defined declarations or computations. Only direct numerical values are allowed. The parameter name will appear in the NSC Editor and the value may be changed as any other parameter value for any object.

Constant

CONSTANT name, value

The CONSTANT command defines the name and numerical value of a constant which can subsequently be used in the place of any numerical value.

Objects

Object commands create new objects from a list of parameters, and possibly references to existing objects.

Cube

CUBE name, xwidth, ywidth, zwidth

Creates a rectangular volume with the specified width in each direction. The origin is at the center of the cube.

Cone

CONE name, r1, r2, L
Creates a cylindrical or cone shaped volume with the front radial size r1, rear radial size r2, and a length L. Either r1 or r2 may be zero, but not both. L may be positive or negative, but not zero. The origin is at the center of face 1 which lies in the XY plane; the length is along the +Z axis.

Cylinder

CYL name, radius, length

Creates a cylinder with the specified radius and length. Length may be positive or negative, but not zero. The origin is at the center of the face which lies in the XY plane; the length is along the Z axis.

Compound Parabolic Concentrator

CPC name, aperture, angle, length, index

Creates a rotationally symmetric compound parabolic concentrator (CPC). The angle is in degrees. If an index greater than 1.0 is provided, then the angle is the refracted angle in the medium of that index.

Rectangular Compound Parabolic Concentrator

CPCR name, aperture_y, angle_y, length, aperture_x, angle_x, index
Creates a rectangular compound parabolic concentrator (CPCR). The angle is in degrees. If an index greater than 1.0 is provided, then the angle is the refracted angle in the medium of that index.

Elliptical Volume 1

ELLIPSE name, rx, ry, L

Creates a volume with elliptical front and back faces with the specified radial size along the X and Y directions, extruded a length L along the Z direction. L may be positive or negative, but not zero. The origin is at the center of the front face.

Elliptical Volume 2

ELLIPTICAL name, front_x, front_y, back_x, back_y, length
Creates a volume with elliptical front and back faces with the specified radial size along the X and Y directions, extruded a length L along the Z direction. L may be positive or negative, but not zero. The origin is at the center of the front face.

Lens

LENS name, r1, k1, radialAperture1, thick, r2, k2, radialAperture2, bevel1, bevel2
Creates a standard lens. The r1 and r2 values are the front and back radii of curvature (use 0 for flat). The k1 and k2 values are the conic constants. The radial clear aperture of the front and back are radialAperture1 and radialAperture2, respectively, and the center to center thickness is given by thick. The origin is at the center of the front face. Optionally the bevel1 and bevel2 values may be added to the radial edge aperture front and back faces.

Polygon

POLYGON name, n, size, depth

Creates an extruded n-sided polygon. The value n must be at least 3. Size is the distance from the center axis to any vertex on the polygon. The front face lies in the XY plane and the depth is the extruded distance along Z. The depth may be positive or negative, but not zero. One vertex of the polygon lies on the +x axis.

Pyramid

PYRAMID name, n, size, depth

Creates an extruded n-sided pyramid. The value n must be at least 3. Size is the distance from the center axis to any vertex on the polygon. The base face lies in the XY plane and the depth is the extruded distance along Z. The depth may be positive or negative, but not zero. One vertex of the polygon lies on the +x axis.

Slot

SLOT name, radius, spacing, depth
Creates a slot with semi-circular sides, a center to center spacing, and a depth in the Z direction. The origin is at the center of the slot.

Sphere

SPHERE name, radius
Creates a sphere with the specified radius. The origin is the center of the sphere.

Spiral

SPIRAL name, radius, R1, R2, turns, pitch

Creates a spiral shape. The spiral axis is along the +Z axis. The spiral "wire" has a circular cross section of radial size "radius". The radius of the windings from the Z axis to the center of the wire is given by R1 at the beginning of the spiral and by R2 at the end of the spiral. The turns value is the number of full windings, and need not be an integer. The pitch value is the change in Z coordinate per turn. The radius parameter should be no larger than one third of the smaller of R1 and R2.

Torus

TORUS name, R, r, start_angle, stop_angle

Creates a circular torus volume. A circle of radius r is placed in the YZ plane a distance R along the +Z axis. The angle 0 corresponds to this position of the circle. The circle is spun about the +Y axis from the start angle to the stop angle to form the torus. The start and stop angles are measured in degrees and must be between -360 and +360.

Sketch Objects

Sketch Object commands can be classified under the object creation commands. Sketch Object commands require a user defined sketch (made in Sketch Mode) associated with each sketch Object command.

Prism/Extrusion
 EXTRUDE name, sketch, draft_angle, length
Creates a volume using the sketch specified in sketchID. The volume is aligned along the z-axis and forms a prism of the given length. The user may also specify a draft angle through the value d_angle. If no draft angle is present, the value is set to 0.0. The length of the prism cannot be 0. The sketch used in the operation must be closed, and curves within the sketch may not overlap.

Pyramid
 PYRAMID_S name, sketchID, length
Creates a pyramidal volume using the sketch specified in sketchID. The volume is aligned along the z-axis and forms a point on the z-axis at the specified length. The length of the pyramid cannot be 0. The sketch used in the operation must be closed, and curves within the sketch may not overlap. The sketch may only use straight line segments.

Revolve
 REVOLVE name, sketchID, start_angle, end_angle
Creates a volume made from the revolution of the sketch specified with sketchID. The sketch is roatated from start_angle to end_angle about the y-axis. The sketch may only on either side of the y-axis, and must be a closed shape. The start_angle and end_angle must be between 0 and 360.

Operations Group

Shapes

Array

ARRAY name, object, nx, ny, nz, dx, dy, dz
Creates an array of the geometry defined by object. The geometry defined by the named object is replicated nx, ny, and nz times with a spacing of dx, dy, and dz. The array is always centered in the coordinate system, whether the n value are even or odd. Note this command creates a new object from object1 and the original object is unaltered. Arrays cannot be created if they consist of over 200 individual objects; This means nx * ny * nz may not exceed 200.

Copy

COPY name, object
Creates a new object by making a copy of an existing object. The original object is unaltered.

Ring

RING name, object, x, y, z, cx, cy, cz, count
Creates a ring array of the geometry defined by object. The ring is centered at the point (x, y, z) and the axis of rotation of the part is given by the direction cosines (cx, cy, cz). The total number of replicated objects is given by count. Before using the RING command, the reference object must be offset from the axis of rotation via a MOVE or POSITION command. Note this command creates a new object from object1 and the original object is unaltered. The parameter "count" may not exceed 200 objects.

Chamfer

 CHAMFER object, width

Adds a chamfer of the specified width to all sharp edges on the object.

Fillet

 FILLET object, radius

Adds a rounded fillet of the specified radius to all sharp edges on the object.

Free

 FREE object

Releases the memory associated with the specified object. This command saves system resources, but is not strictly required as objects are automatically freed at the end of the script.

Scale

 SCALE object, sx, sy, sz

Scales the named object by the specified factors in the X, Y, and Z directions. The scale values must not be zero, but may be positive or negative. If negative, the object is reversed in the corresponding direction.

Color

 COLOR object, R,G,B

Colors an object created by an object creation command with the RGB values specified. Each value must be between 0-255.

Translate

Move

MOVE object, dx, dy, dz

Moves the named object by the specified translation in x, y, and z directions.

Rotate

ROTATE object, angle, cx, cy, cz
Rotates the named object by the specified angle in degrees about the arbitrary axis defined by the x, y, and z direction cosines.

Mirror

MIRROR object, x, y, z, cx, cy, cz
Mirror copies the named object, and reflects the copy relative to a plane defined by the specified (x, y, z) point and normal vector with direction cosines (cx, cy, cz). The original and the copy are then merged to be a single object, whether or not they touch. See also REFLECT, which reflects an object without making a copy.

Reflect

REFLECT object, x, y, z, cx, cy, cz
Reflects the named object relative to a plane defined by the specified (x, y, z) point and normal vector with direction cosines (cx, cy, cz). See also MIRROR, which reflects an object and merges it with the unreflected original object.

Position

POSITION object, x, y, z, cx, cy, cz

Moves and rotates the object to a new coordinate system centered on the specified (x, y, z) point with the local +Z axis pointing along the direction cosines (cx, cy, cz). The original origin is assumed to lie on the new ZY plane.

Boolean

Union

UNION name, object1, object2

Creates a new object that is the Boolean union of object1 and object2.

Difference

DIFFERENCE name, object1, object2
Create a new object that is the Boolean difference between object1 and object2. This command is used to remove from object1 the volume represented by object2.

Overlap

OVERLAP name, object1, object2
Create a new object that is the Boolean overlap (logical AND) of object1 and object2. This command is used to select only the volume that is shared by both object1 and object2.

Ray

RAY data, x, y, z, cx, cy, cz, object
The RAY command defines a ray at the specified x, y, and z point, with the specified cx,cy,and cz direction cosines, and then traces the ray to the named object. If the ray intercepts the object, the point of interception is stored in the named data value. The named data value contains three values: the x, y, and z coordinate of the intercept point. These values may be used as constants in subsequent script commands.

Example
 Placing a small sphere on the top of a cube
This script defines a cube, then a ray starting above the cube intercepts the top face of the cube, and defines a point where a sphere is placed.

CUBE box, 1, 1, 1
RAY point, 0, 10, 0, 0, -1, 0, box
SPHERE ball, 0.3
MOVE ball, point.X, point.Y, point.Z
UNION final, box, ball

Note the individual coordinates of the point of intersection

Math Syntax

The Zemax Part Designer is also capable of performing simple mathematical calculations as a substitute for data values within a script. The following operations may be used:

+, -: Addition and subtraction

*, /: Multiplication and division

(): All operations and values may be wrapped within parenthesis

SQRT data_value: the Square root command evaluates the square root of the data value following the command. The input cannot be less than or equal to 0.

SINE, COSI, TANG, ATAN, ASIN, ACOS: The Sine, Cosine, Tangent, Arc Tangent, Arc Sine, and Arc Cosine commands, respectively. These commands are followed by a data value given in radians.

The PARAMETER command is not compatible with these operations.

Sketch Group
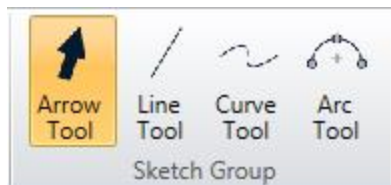
Arrow Tool
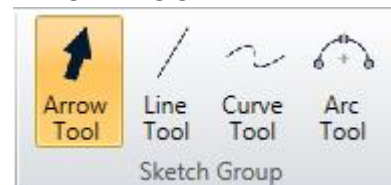
The Arrow Tool enables a user to select and manipulate points that have already been plotted. Selected points appear yellow to the user. In general, selected points can be dragged using the left mouse button and can be used in conjunction with the various ZPD sketch hot keys such as 'g' (snap to grid line), or 'v'(snap to integer). Selected points may also be deleted using the delete key. Deleting a point creates a Bezier Curve between the points before and after the deleted point.

Points at the ends of a Bezier Curve can be used to adjust the "tangency" of a curve. The appearance of a Bezier Segment is dictated by the start and end tangent points of the curve. These tangent vectors are anchored at end points of a curve and point out into 2D space. The magnitude and direction of these vectors determine the shape of the curve. Tangent vectors appear as dotted lines in the Editor window.

Typically one point controls the end tangent point of the previous segment while the other point controls the start tangent point of the next segment. Modifying the magnitude or changing the direction of a vector will affect segment's shape in varying ways.

To adjust the magnitude and direction of a tangent vector, a control point corresponding to either a start or end point of a Bezier Segment must be selected. 1-2 dotted lines should appear, each representing either the start or end vector of the next or previous curve. Right clicking and dragging either point at the end of the dotted line will move it to the mouse cursor's location, changing the curve's appearance in real time.

The arrow tool can also be used to select highlighted arc segments. Upon selecting an arc segment, the arc manipulation panel will appear and replace the current point manipulation panel. These panels are discussed later in the "Exposing and Parameterizing Sketches within Zemax" section of the documentation.

To deselect a point, use the arrow tool and click in a blank space.

Line Tool
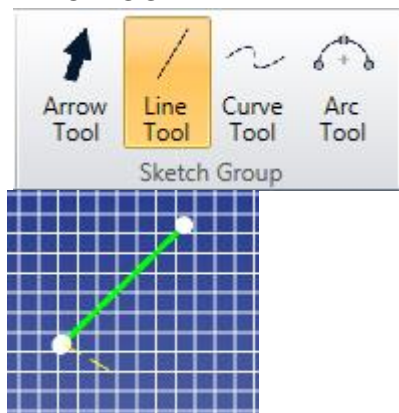
The line tool creates a basic straight line segment. Clicking the left mouse button within the sketch places the first point. A subsequent click will draw a line segment from the first point to the last mouse click.

The user cannot adjust the tangent vectors for straight line segments.

Curve Tool
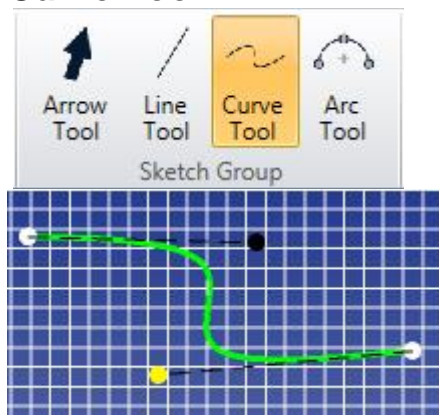
To create a curve, select the Bezier tool and click once to plot a starting point. A subsequent click will draw a curve from the first point to the last mouse click. Clicking and releasing the left mouse button will automatically adjust the start tangency to align with a prior curve or line segment's end tangency, thus creating a continuous or smooth connection between the old and new segments. Clicking and holding the left mouse button (as opposed to clicking and releasing the left mouse button) will place an end point in a similar fashion, however, dragging the mouse will place the end tangent point of the newly created curve at the location of the mouse cursor.

When a curve has been closed, new control points may be added along the existing curve by selecting the line tool or the Bezier tool and clicking anywhere along the curve. Each left mouse click will place a new point at the mouse cursor, separating the clicked line or curve segment into 2 adjustable Bezier segments or straight line segments, depending on the type of segment being altered.

Arc Tool
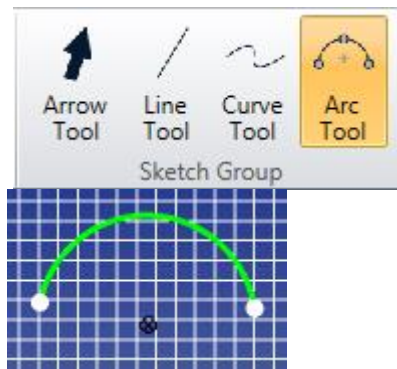
The Arc Tool is used to create an arc segment that is limited to one half-circle. Creating an arc requires three simple steps:

1)   Select the Arc Tool from the toolbox and plot a point. If you are continuing a sketch from an existing point, a new point does not need to be plotted to start an arc.

2)   Click once to place the location of the end point

3)   Adjust the preview curve by moving the mouse cursor. The final click will plot the arc and produce a new control point to draw from at the end point.

The start and end points of the arc are adjustable using the Arrow Tool.

Search Bar
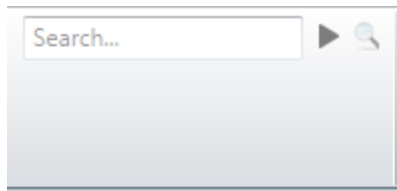
This search feature helps users locate Part Designer tools in the OpticStudio Part Designer tab.

Type a keyword into the search bar to find related analyses or tools.

Button Functions

Run Feature (Play icon) Opens/runs the selected Part Designer tool or analysis.

Highlight Feature (Magnifying glass icon) Highlights the location of the selected tool or analysis in the OpticStudio UI.

The Programming Tab

This tab provides access to OpticStudio's Programming features.
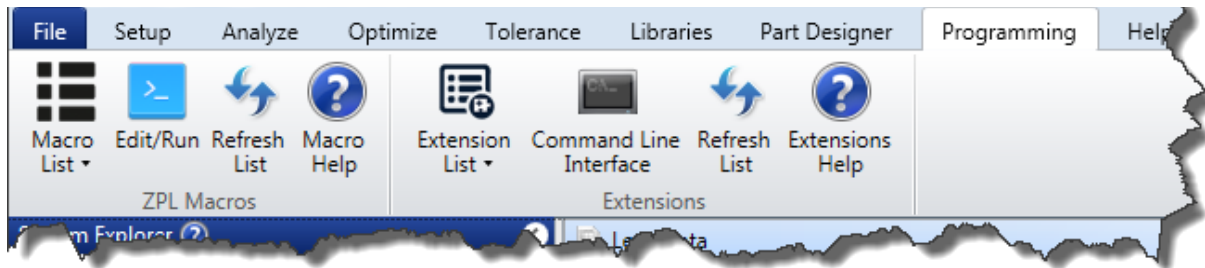
ZPL Macros Group

ZPL Macros is a section of the Programming Tab.
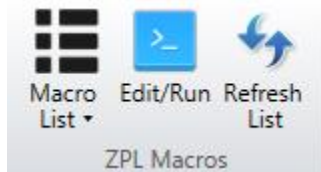
For more information, please see the section About the ZPL

Macro List

The Macro List button is found in the ZPL Macros section of the Programming Tab.
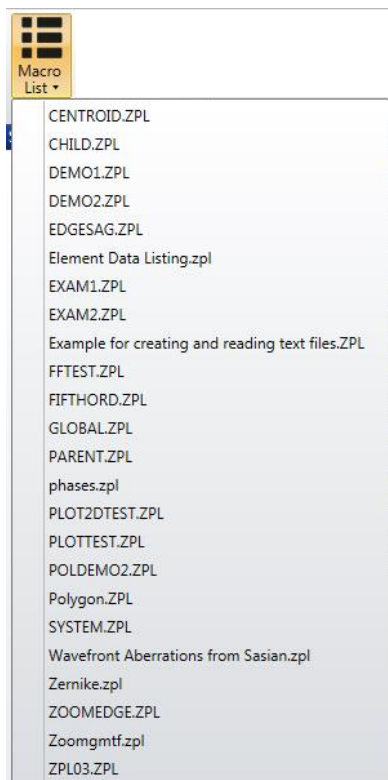
This feature shows a drop-down list of all ZPL Macros in the current macros folder (see "Folders").

By clicking on the macro name, it immediately executes.

Edit/Run

The Edit/Run macro tool is found in the ZPL Macros section of the Programming Tab.
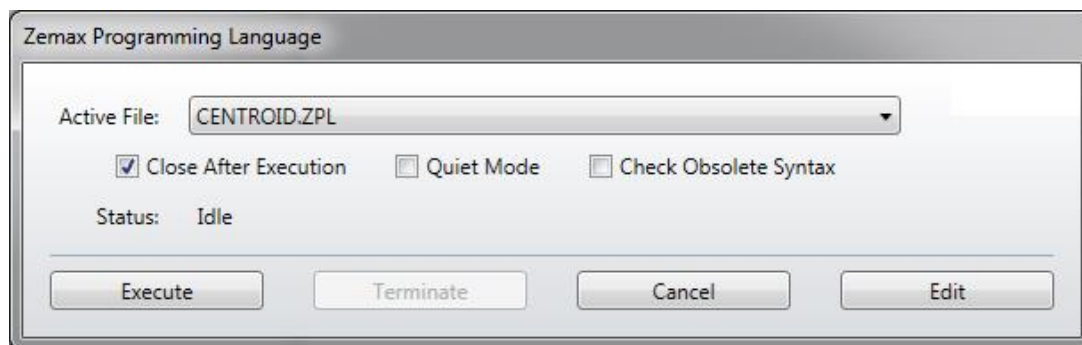
This tool edits or runs the specified macro.

To run a ZPL macro, select the Edit/Run button in the ZPL Macros section of the Programming Tab. The ZPL control dialog box will appear with the following options:

Active File A drop-down list of macros available. All the listed macros are text files ending in the extension .ZPL. The files must be in the folder for ZPL Macros; see "Folders". Macros may also be placed in any subfolder within the macros folder.

Close After Execution If checked, the ZPL control dialog box will automatically close after the macro execution.

Quiet Mode If checked, the default output text window will not be shown. This is useful for graphics macros that do not generate useful text.

Check Obsolete Syntax If checked, Zemax will test the macro for use of obsolete syntax.

Status During execution of the macro, Zemax will use this area to print a status message stating the line number of the macro being executed. The status message is updated every quarter second.

Execute Click to begin execution of the selected macro.

Terminate The terminate button will stop execution of the macro currently running.

Cancel The cancel button terminates the current macro if one is running. If no macro is running, cancel closes the ZPL control dialog box.

Edit The edit button invokes a text editor in OpticStudio. The editor can be used to modify or rename the macro.

Select the macro to run from the "Active File" list, and then click on Execute.

Zemax will begin running the macro. The entire macro will run, and any text output from PRINT commands or error messages will be displayed in a window. The display of this output window can be suppressed using the CLOSEWINDOW keyword.

Refresh List

The Refresh macro list button is found in the ZPL Macros section of the Programming Tab.
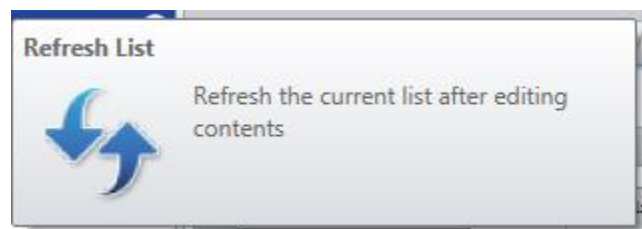
This tool updates the current macro list; which may be required if any macros have been added or deleted since Zemax was started or the last time the list was refreshed.

About the ZPL

This section is about using the Zemax Programming Language (ZPL). If you are viewing this in an HTML dialog, expand the header in Contents tab to view the subsections.

Introduction

The Zemax Programming Language (ZPL) is a macro language specifically designed for use with Zemax. ZPL offers the power of user-extensibility. This means that if you need a particular calculation or graphical display which is not "built in", you can write your own ZPL macro to do the job.

ZPL is similar to the BASIC programming language, except not all BASIC constructs and keywords are supported, and capabilities and functions unique to ray tracing have been added. ZPL is easy to use, and this section will give you instructions and examples to get you started. For an alternate programming capability see "About Extensions".

ZPL is a powerful programming language. While it is easy to use, the user is responsible for error-checking, debugging code logic and for good programming practice. For this reason, technical support on macro writing is restricted to

ensuring that all ZPL functions and keywords work as documented: we cannot advise on how to perform detailed calculations as part of technical support. If you need a Zemax macro, and do not possess the desire or ability to write it yourself, please feel free to contact Zemax Technical Support for a quote on developing a custom program to meet your requirements. We have considerable experience in developing these types of programs, and can generally write macros at very competitive rates on short notice.

Creating ZPL Macros

A ZPL macro consists of a series of commands which are stored in a text file. The commands are either assignments, keywords, or comments. See "ZEMAX PROGRAMMING LANGUAGE" for more information.

To create a ZPL macro, it is probably easiest to start with an existing macro that performs a task similar to the one you want to achieve. If you are attempting to write your first ZPL macro, you may want to read the example sections at the end of this help file. Some example ZPL macros can also be found in the <data>\Macros folder installed with the Zemax program.

Use any text editor to create the ZPL file (such as the NOTEPAD editor). The file may have any name but must end in the .ZPL extension. The file must be placed in the ZPL Folder, which by default is <data>\Macros. To change this folder see "Folders".

There is a limit to the allowed complexity of any single line in a ZPL macro. If the "line too long" error occurs, try breaking the line into several smaller lines.

An Overview of ZPL

A ZPL macro consists of a series of commands which are stored in a text file. The commands are either assignments, keywords, or comments. Assignments may be for either numeric or string (textual) data. Both assignments and keywords may accept expressions as arguments, although the syntax is slightly different as described below.

Assignments

The general syntax for an assignment is

```
variable = (expression)
```

The (expression) may consist of an explicit value, such as 5, or a variable name containing some preassigned value, or a complex mathematical expression involving functions, constants, and variables. In all cases, the expression on the right side of the equal sign is evaluated, and the result is assigned to the variable designated on the left.

The simplest form of an assignment is where the expression is a fixed value, such as:

```
x = 5
```

There are several important things to note in this command. First, no declaration of variables is required. This means "x", which is called a variable, did not need to exist before the value of 5 was assigned to it. If "x" had already been assigned a value, it would now be reassigned. Second, no special symbol is required to terminate a command, such as ";" in C. Because of this, each ZPL command must be on a line by itself.

Here are some examples of assignments with expressions:

```
x = SQRT(5)
y = SINE(x)
z = SQRT(x+5*(7-x))
```

The functions SQRT (square root) and SINE (sine) are built in to ZPL. There are many of these functions, all of which are defined in "Numeric functions". Note that ZPL is not case-sensitive; SQRT() and sqrt() are the same function. This documentation will use the convention of capital letters for functions and keywords, and lower case for everything else.

There are also string assignments as described in "String variables".

Keywords

The general syntax for a keyword is

```
KEYWORD argument1, argument2, argument3...
```

Some keywords have no arguments, others have many. Arguments may be either numeric expressions or string constants or string variables. Some keywords accept a mixture of numeric and string arguments. One example of a keyword is PRINT. The PRINT keyword is followed by a list of items, separated by commas, to be printed. For example, the ZPL commands

```
x = 3 y = 4
z = SQRT( x * x + y * y ) PRINT "The hypotenuse is ",z
```

will print the following:

```
The hypotenuse is 5.0000
```

Note that ZPL enforces operator precedence. ZPL uses the following precedence from highest to lowest: Parentheses, functions (such as SQRT), logical operators (such as ==), multiplication and division, and then addition and subtraction.
There are many keywords, all are described in "Keywords".

Comments

There are 3 ways to add comments to a ZPL macro: by starting a line with the REM keyword, by starting a line with the "!" symbol, or by placing the "#" symbol anywhere on the line as long as the # symbol is not inside a string. Blank lines are also allowed anywhere in the macro. Here are 3 examples of comments:

```
REM this is a remark
! This is also a remark
x = 5 # The # symbol allows comments on the same line as a valid command
```

Comments make macros easier to understand and modify, and have no effect on performance.

Creating Graphics

There are a number of low-level functions for generating graphics, including GRAPHICS, GTEXT, GLEN-SNAME, and others. For details see "GRAPHICS". An easier method of generating typical Zemax style line graphs and 2D graphs graphics is available using the PLOT and PLOT2D keywords; see "PLOT" and "PLOT2D".

Numeric variables

Variables provide temporary storage for numerical quantities whose exact value may not be known when the macro is written, but is defined when the macro is run. Zemax performs most of the work for you when you need a new variable. For example, the command

```
x = 5
```

will cause Zemax to allocate memory for the new variable "x" and keep track of the value associated with it. Once the variable is defined, it may be used in any subsequent expression. There are a few rules regarding ZPL variables:

Variable names must not contain any "special" characters that ZPL uses for logical operations or delimiting such as (,), =, +, -, *, /, !, >, <, ^, &, |, #, ", and the space character.

A variable cannot take on the same name as a keyword or function, such as THIC or RAYX. Since ZPL is not case-sensitive, you cannot use rayX or Thic to avoid this rule.

Each variable name is limited to 28 characters.

All ZPL numeric variables are stored internally as 64-bit double precision numbers.

Array variables

Array variables are single- or multi-dimensioned arrays of double precision or integer values. Unlike (scalar) numeric variables, array variables must be declared prior to their use. The declaration syntax is

```
DECLARE name, type, num_dimensions, dimension1 [, dimension 2 [, dimension 3 [,
dimension 4] etc...]]
```

The name may be any legal variable name as described in the previous section. The type must be either DOUBLE or INTEGER; this value indicates the type of array variable. The integer value num_dimensions defines the number of dimensions of the array (not the size), and must be between 1 and 4, inclusive. The integers dimension1, dimension2, etc., define the size of the array in that dimension. Note that array variables start at index 1, and thus an array of size 10 has valid indices from 1 to 10.

Array variables may be defined anywhere inside the macro, they need not be declared at the beginning of the macro. To release the memory associated with an array variable, use the RELEASE keyword. The syntax is

```
RELEASE name
```

The RELEASE keyword is optional, as the memory associated with the declared variable is automatically released when the macro terminates. However the RELEASE keyword is useful for conserving memory if large arrays are only needed during a portion of the macro execution.

Array variables are assigned values using the following syntax:

```
name ( index1, index2, ... ) = value
```

The values stored in the array may be retrieved with the same basic syntax:

```
value = name ( index1, index2, ...)
```

The following sample code declares a two-dimensional array variable, assigns a value to each element, prints the values, and then releases the memory for the array:

```
DECLARE Z, DOUBLE, 2, 5, 5
FOR i, 1, 5, 1
     FOR j, 1, 5, 1
            Z(i, j) = i + j
     NEXT j
NEXT i

FORMAT 8.0 k = 0
FOR i, 1, 5, 1
     FOR j, 1, 5, 1
            PRINT k, i, j, Z(i,j)
     k = k + 1
     NEXT j
NEXT i

RELEASE Z
```

Numeric operations

ZPL macros support basic numeric operations such as add, subtract, multiply, and divide. The syntax for each is shown below.

```
x = y + z
x = y - z
x = y * z
```

```
x = y / z
```

All other operations are supported only through the use of numeric functions or numeric logical operators, both described in subsequent sections.

Numeric logical operators

Logical operators are used to construct complex commands which ultimately evaluate to one or zero. Most logical operations take the form (left_expression) (operator) (right_expression), similar to mathematical expres- sions such as 1 + 2. The exception is the not operator "!" which takes only a single argument, of the form ! (right_expression). The logical operators use the convention that zero is "false" and any non-zero value is "true". The not operator returns 1 (true) if the (right_expression) is 0 (false) and returns 0 (false) if (right_expression) is non-zero (true). One common use of the not operator is in IF commands such as:

```
IF !x THEN PRINT "x is zero."
```

The other logical operators can be also be used as part of the argument in IF commands. For example, an IF command may contain two conditions which must both be true for the THEN command to be executed:

```
IF ( x > 1 ) & ( y < 2 ) THEN PRINT "Both conditions are true."
```

These two conditions are related by an "and" expression denoted by &. Note the parentheses are used to force precedence. ZPL supports the logical operators described in the following table.

NUMERIC LOGICAL OPERATORS

Logical	Description
&	And, returns 1 only if both expressions are non-zero.
|	Or, returns 1 if at least one expression is non-zero.
^	Xor, returns 1 if only one expression is non-zero.
!	Not, returns 0 if (right_expression) is non-zero, else returns 1.
==	Equality, returns 1 if expressions are equal.
>	Greater than, returns 1 if left_expression is greater than right_expression.
<	Less than, returns 1 if left_expression is less than right_expression.
>=	Greater than or equal to, returns 1 if left_expression is greater than or equal to right_expression.
<=	Less than or equal to, returns 1 if left_expression is less than or equal to right_expression.
!=	Inequality, returns 1 if expressions are unequal.

String variables

ZPL supports string variables and basic string operations. String variables can hold a maximum of 260 characters. String variables do not need to be declared, but can be created at any time using a defining assignment command such as:

```
newstring$ = "Here is the new string"
```

Note that string variables are distinguished from numeric variables by the presence of the $ character at the end of the string.

There are also string functions which can be used to extract text data, such as

```
title$ = $LENSNAME()
```

Note the function $LENSTITLE() starts with the $ character. This identifies the function as returning a string result.

String operations

String variables can be concatenated using the + operator. The syntax is:

```
C$ = A$ + B$
```

Concatenation can also include constant strings:

```
total$ = "A$ is " + A$ + " and B$ is " + B$
```

The string functions can be used in a defining command such as

```
this$ = "Here is the lens title: " + $LENSNAME()
```

String variables are printed just like other strings:

```
PRINT "Here is A$: ", A$
```

Note that the PRINT function can only print single string variables; there is no support for the concatenation operand or string functions inside print commands. The correct procedure is to concatenate the strings into a new string and then print the new string:

```
A$ = B$ + C$
PRINT A$
```

Alternatively, the comma acts as a concatenation operand:

```
PRINT A$, B$, C$
```

String functions cannot be printed directly like this:

```
PRINT $LENSNAME() ! NOT CORRECT !!!
```

Instead, the correct procedure is to assign the function result to a variable and then print the variable:

```
Z$ = $LENSNAME()
PRINT Z$
```

String logical operators

String logical operators are very similar to the numeric logical operators defined in a previous section. The key difference is that the expressions being compared are strings rather than numbers. The supported string logical operators are defined in the following table.

STRING LOGICAL OPERATORS

Logical	Description
$==	Equality, returns 1 if left_string and right_string are identical.
$>	Greater than, returns 1 if left_string is greater than right_string.
$<	Less than, returns 1 if left_string is less than right_string.
$>=	Greater than or equal to, returns 1 if left_string is greater than or identical to right_string.
$<=	Less than or equal to, returns 1 if left_string is less than or identical to right_string.
$!=	Inequality, returns 1 if left_string and right_string are not identical.

For example, an IF command may compare strings as follows:

```
A$ = "TEST"
BS = "TEST"
IF (A$ $== B$) THEN PRINT "Strings are identical."
```

Numeric functions

Numeric functions can be used on the right hand side of a numeric variable assignment, and in expressions which are arguments to keywords. These functions may require no arguments, one argument, or multiple arguments. All functions return a single value. Certain functions, such as PWAV() (primary wavelength) , return a value which does not depend upon the argument, and therefore it is not required to provide one. The parentheses however, are still required.

In the following table, all ZPL functions are listed. If the syntax is given as FUNC(), then no arguments are required. FUNC(x) indicates one argument is required, FUNC(x,y) indicates two arguments, etc.

ZPL FUNCTION LISTING

Function	Argument	Return value
ABSO(x)	Numeric expression	The absolute value of the expression.
ACOS(x)	Numeric expression	Arc cosine in radians.
APMN(x)	Valid surface number	The minimum radius value. For spider apertures, this is the width of the arms. For rectangular and elliptical apertures, it is the x-half width of the aperture.
APMX(x)	Valid surface number	The maximum radius value. For spider apertures, this is the number of arms. For rectangular and elliptical apertures, it is the y-half width of the aperture.

APOI(px, py)	Normalized pupil coordinates	Ray intensity apodization factor.
APXD(x)	Valid surface number	The aperture x-decenter value.
APYD(x)	Valid surface number	The aperture y-decenter value.
APTP(x)	Valid surface number	An integer code describing the aperture type on the specified surface.
ASIN(x)	Numeric expression	Arc sine in radians.
ASPR()	(null)	Aspect ratio of the current graphics device.
ATAN(x)	Numeric expression	Arc tangent in radians.
ATYP()	(null)	System aperture type code: 0 for EPD, 1 for F/#, 2 for NA, and 3 for float by stop size.
AVAL()	(null)	System aperture value.
CALD(i)	index	Returns a numeric value from the CALLMACRO buffer at the specified index. See "Calling a Macro from within a Macro"
CONF()	(null)	Returns the current configuration number, which is between 1 and the number of configurations, inclusive.
CONI(x)	Valid surface number	Conic constant of the surface.
COSI(x)	Numeric expression in radians	Cosine of the expression.
CURV(x)	Valid surface number	Curvature of the surface.
EDGE(x)	Valid surface number	Edge thickness at the semi-diameter of that surface.
EOFF()	(null)	End of file flag. Returns 1 if the end of file has been reached, otherwise returns 0. Only valid after execution of READ keyword.
ETIM()	(null)	The elapsed time in seconds since last TIMER.
EXPE(x)	Numeric expression	e to the power of the expression.
EXPT(x)	Numeric expression	10 to the power of the expression.
FICL(vec#)	Vector #, between 1 and 4 inclusive..	The fiber coupling efficiency. See "Using the FICL() function".

FLDX(x)	Valid field number	X-angle or height of specified field.
FLDY(x)	Valid field number	Y-angle or height of specified field.
FTYP()	(null)	Returns 0 if the current field type is angles in degrees, 1 if object height, 2 if paraxial image height, and 3 if real image height. Heights are always in lens units (see UNIT).
FVAN(x)	Valid field number	Vignetting angle of specified field.
FVCX(x), FVCY(x)	Valid field number	Vignetting compression x or y of specified field.
FVDX(x), FVDY(x)	Valid field number	Vignetting decenter x or y of specified field.
FWGT(x)	Valid field number	Field weighting of specified field.
GABB(x)	Valid surface number	The glass catalog Abbe number of the glass for the specified surface.
GAUS(x)	Standard deviation	Returns a random value with a Gaussian distribution, zero mean, and the specified standard deviation.
GETT(window, line, column)	The window number, line number, and column number of the numerical value to extract. Each Columns are delimited by spaces.	Extracts a numerical value from any open text window. This feature allows computations using any value Zemax can display in a window.
GIND(x)	Valid surface number	The glass catalog d-light index of the glass for the specified surface.
GLCA(x)	Valid surface number	Global vertex x vector of the specified surface.
GLCB(x)	Valid surface number	Global vertex y vector of the specified surface.
GLCC(x)	Valid surface number	Global vertex z vector of the specified surface.
GLCM(surf, item)	Surf must be a valid surface number, item is an integer between 1 and 12.	For item equal to 1-9, the return value is R11, R12, R13, R21, R22, R23, R31, R32, or R33. For item equal to 10-12, the return value is the x, y, or z component of the global offset vector. See "Global Coordinate Reference Surface".
GLCX(x)	Valid surface number	Global vertex x-coordinate of the specified surface.
GLCY(x)	Valid surface number	Global vertex y-coordinate of the specified

		surface.
GLCZ(x)	Valid surface number	Global vertex z-coordinate of the specified surface.
GNUM(A$)	Any string variable name	If the string A$ is the name of a valid glass, such as BK7, then GNUM returns the number of the glass in the glass catalog. The glass number can subsequent l y be used by SETSURFACEPROPERTY to set the glass type on a surface. If A$ does not correspond to any glass in the catalog, GNUM returns 0. GNUM returns -1 if the string is "MIRROR".
GPAR(x)	Valid surface number	The glass catalog partial dispersion coefficient of the glass for the specified surface.
GRIN(s, w, x, y, z)	Surface #, wavelength #, x, y, and z coordinates	Returns the index of refraction at the specified x, y, z position on surface s at wavelength number w. Works for gradient and non-gradient media.
GTEM(code)	For checkbox values, code is 0 for Use Glass Substitution Template, 1 for Exclude Glasses With Incomplete Data, 11 for Standard, 12 for Preferred, 13 for Obsolete, 14 for Special, and 21-26 for Use Relative Cost, CR, FR, SR, AR, and PR, respectively. For numerical values, code is 31-36 for Relative Cost, CR, FR, SR, AR, and PR, respectively.	Returns data from the Glass Substitution Template. Checkboxes return zero for false (unchecked) and any non-zero value for true (checked).
IMAE(seed)	The seed value	Returns the geometric image analysis efficiency. See the discussion "Geometric Image Analysis". If seed is zero, the same random numbers will be used on every call to IMAE. If seed is not zero, then every call to IMAE will use unique random numbers.
INDX(surface)	Valid surface number	Index of refraction at the primary wavelength. See ISMS.
INTE(x)	Numeric expression	Returns the largest integer not greater than the argument.
ISMS(surface)	Valid surface number	If the surface is an odd mirror, or follows an odd mirror but is not a mirror, the return value will be one, otherwise the return value is 0.

LOGE(x)	Positive numeric expression	Log base e of the expression.
LOGT(x)	Positive numeric expression	Log base ten of the expression.
LOST(code)	Code is 1 to return lost energy due to errors, or 2 for lost energy due to thresholds.	The lost energy following the most recent NSTR trace.
MAGN(x,y)	x and y are any real numbers	Computes the square root of x squared plus y squared.
MAXF()	(null)	The maximum radial field angle in degrees, or radial object or image height in lens units. Which value is returned depends upon if the fields are defined by angle, object height, or image height.
MAXG()	(null)	The number of glasses currently loaded.
MCON(row, config, data)	Row (operand number) ,Configuration number, and data value to extract from the Multi-Configuration Editor.	Extracts data from any row and configuration of the Multi-Configuration Editor. This function is similar to MCOP with extended capabilities for extracting data.
		If the row and config number are both zero, MCON returns either the number of operands, the number of configurations, or the active configuration number for data = 0, 1, and 2, respectively.
		If the row number is between 1 and the number of multi-config operands, and the config number is zero, MCON returns the operand type, integer 1, integer 2, integer 3, and string flag for that specified row, for data = 0 through 4 respectively. The 3 integer values are used for various purposes for different operands, such as surface and wavelength numbers. The string flag is 1 if the operand data is a string value, such as a glass name, or 0 for numerical data.
		If the row number is between 1 and the number of multi-config operands, and the config number is valid, MCON returns either the numerical value or the string data for that operand.
		Note that all string data returned by MCON must be extracted with the $buffer command after the call to MCON. For example, the following code sample will place the name of the operand on row 1 in a$:
		dummy = MCON(1, 0, 0)

		a$ = $buffer()
		See also keyword SETMCOPERAND.
MCOP(row, config)	Row (operand number) and configuration of the Multi - Configuration Editor.	Extracts data from any row and configuration of the Multi-Configuration Editor. A configuration number of zero indicates the current configuration. See al so keyword SETMCOPERAND.
MFCN()	(null)	MFCN updates the lens, validates the merit function, updates the merit function, then returns the current merit function value. See OPER.
NCON()	(null)	Returns the number of configurations.
NFLD()	(null)	The number of defined fields.
NOBJ(surface)	Valid non-sequential surface number	The number of defined objects in the non-sequential surface.
NPAR(surf, object, param)	surf is the surface number, object is the object number, and param is the parameter number of the value to return.	Returns a value from the parameter columns of the non-sequential components editor. See "SETNSCPARAMETER".
NPOS(surf, object, code)	surf is the surface number, object is the object number, and code is 1-6 for reference object, x, y, z, tilt x, tilt y, and tilt z position to return, respectively.	Returns a value from the position columns of the non-sequential components editor. See "SETNSCPOSITION".
NPRO(surf, object, code, face)	surf is the surface number, object is the object number, code values are as def ined for the SETNSCPROPERTY keyword, and face is the face number	Returns a numeric or string value from the property pages of objects defined in the non-sequential components editor. For information on the code values, see "SETNSCPROPERTY". String values may be extracted using the $buffer() function after calling this function with a code that returns string data. For example, to extract the comment column of object 5 from NSC surface 2 use these two commands: dummy = NPRO(2, 5, 1, 0) a$ = $buffer() The value dummy may be ignored. The string variable a$ will contain the object comment.
NSDC(surf, object, pixel, data)	The surf value indicates the surface number of non-sequential group. The object number indicates the	If the object number corresponds to a detector rectangle object, then the coherent intensity data from the specified pixel is returned. Coherent intensity consists of 4 numbers which are each

	desired detector. Pixel indicates which pixel value is returned. Data is 0 for real, 1 for imaginary, 2 for the amplitude, and 3 for the coherent intensity. See also "NSTR"	summed over all incident rays: the real part, the imaginary part, the amplitude, and the coherent intensity. If the pixel number is zero, then the sum of the data for all pixels for that detector object is returned.
NSDD(surf, object, pixel, data)	The surf value indicates the surface number of non-sequential group. The object number indicates the desired detector. See the discussion to the right for details on the pixel and data arguments. See also "NSTR".	If the object number is zero, then all detectors are cleared and the function returns zero. If the object number is less than zero, then the detector defined by the absolute value of the object number is cleared and the function returns zero. If the object number corresponds to a detector rectangle, surface, or volume object, then the incoherent intensity data from the specified pixel is returned. For a full discussion of the pixel and data arguments, see "NSDD". This ZPL function supports the same pixel and data arguments as the NSDD optimization operand. This function does not support specifying a value for the # Ignored argument provided by the NSDD optimization operand, and assumes that the value for # Ignored is always zero.
NSDE(surf, object, pixel, angle, data)	The surf value indicates the surface number of non-sequential group. The object number indicates the desired detector. See the discussion to the right for details on the pixel, angle, and data arguments. See also "NSTR".	If the object number is zero, then all detectors are cleared and the function returns zero. If the object number is less than zero, then the detector defined by the absolute value of the object number is cleared and the function returns zero. If the object number corresponds to a detector color object, then the data from the specified pixel is returned. For a full discussion of the pixel, angle, and data arguments, see "NSDE". This ZPL function supports the same pixel, angle, and data arguments as the NSDE optimization operand. This function does not support specifying a value for the # Ignored argument provided by the NSDD optimization operand, and assumes that the value for # Ignored is always zero.
NSDP(surf, object, pixel, data)	The surf value indicates the surface number of non-sequential group. The object number indicates the desired detector.	If the object number is zero, then all detectors are cleared and the function returns zero. If the object number is less than zero, then the detector defined by the absolute value of the object number is cleared and the function returns zero.

	See the discussion to the right for details on the pixel and data arguments. See also "NSTR".	If the object number corresponds to a detector polar object, then the data from the specified pixel is returned.
		For a full discussion of the pixel and data arguments, see "NSDP"This ZPL function supports the same pixel and data arguments as the NSDP optimization operand.
NSUR()	(null)	The number of defined surfaces.
NWAV()	(null)	The number of defined wavelengths.
OBJC(A$)	Any string variable name	Object with comment. Returns the first non-sequential object number where the comment matches the string A$. The comparison is case insensitive. If no NSC object has the matching comment the function returns -1.
OCOD(A$)	Any string variable or literal that is the name of a Zemax optimization operand.	The optimization operand code number used by the OPEV function.
ONUM(A$)	Any string variable name	If the string A$ is the name of a valid optimization operand, such as EFFL, then ONUM returns the id number of the operand.
		The operand id number can subsequently be used by SETOPERAND to set the operand type in the merit function editor. If A$ does not correspond to any operand name, ONUM returns 0.
OPDC()	(null)	The optical path difference. Valid only after a RAYTRACE call. OPDC will not return valid data if the chief ray cannot be traced. RAYTRACEX does not support OPDC.
OPER(row, col)	Row (operand number) and column (data type) of the Merit Function Editor.	Extracts data from any row and column of the Merit Function Editor. The row is the same as the operand number; the column is 1 for type, 2 for int1, 3 for int2, 4-7 for data1-data4, 8 for target, 9 for weight, 10 for value, 11 for percent contribution, and 12-13 for data5-data6.
		To obtain the comment from a BLNK operand, use column number 2. The comment is then placed in the string buffer. For example, to obtain the comment from row 1:
		dummy = OPER(1,2)
		a$ = $buffer()

		The value in the dummy variable may be ignored, and a$ will then contain the comment string.
		OPER does not update the lens or the merit function, it returns the current data. See also OPEV, MFCN and keyword SETOPERAND.
OPEV(code, int1, int2, data1, data2, data3, data4)	Code is the optimization operand code (see function OCOD), and int1-int2, and data1-data4 are the defining values for the operand. See "Optimization operands".	Computes the same value as any optimization operand would, without the need to add the operand to the merit function. This is useful for computing numbers already available from the optimization operands. For example, to compute the EFL from the EFFL operand use this code: C = OCOD("EFFL") E = OPEV(C, 0, 1, 0, 0, 0, 0) See also OPER, OCOD, MFCN and keyword SETOPERAND. Some optimization operands, such as OPGT, only make sense when taken in the context of other operands. OPEV will only work for operands that do not depend upon the presence of prior operands. See also OPEW below.
OPEW(code, int1, int2, data1, data2, data3, data4, data5, data6)	Code is the optimization operand code (see function OCOD), and int1-int2, and data1-data6 are the defining values for the operand. See "Optimization operands".	This function is very similar to OPEV, the key difference being that OPEW supports two additional arguments. Some optimization operands use up to 6 data values rather than 4.
OPTH(x)	Valid surface number	The optical path length along the ray to the specified surface. Unlike RAYT and RAYO, OPTH considers the phase added by diffractive surfaces such as gratings, holograms, and binary optics. Valid only after a RAYTRACE call. OPTH will not return valid data if the chief ray cannot be traced. RAYTRACEX does not support OPTH.
PARM(n,s)	Valid parameter number and surface number	Parameter "n" of surface "s".
PIXX(A$)	A string variable containing the full path to a BMP, JPG or PNG graphic file.	The number of X-direction pixels in the graphic.
PIXY(A$)	A string variable containing the full path to a BMP, JPG or PNG graphic file.	The number of Y-direction pixels in the graphic.

PMOD()	(null)	0 if paraxial mode is off, else 1.
POWR(x,y)	x and y are any numbers	Computes the absolute value of x to the power of y.
PVHX()	(null)	Data1 parameter from the ZPLM optimization operand.
PVHY()	(null)	Data2 parameter from the ZPLM optimization operand.
PVPX()	(null)	Data3 parameter from the ZPLM optimization operand.
PVPY()	(null)	Data4 parameter from the ZPLM optimization operand.
PWAV()	(null)	The primary wavelength number.
RADI(x)	Valid surface number	Radius of curvature of surface. If the surface has an infinite radius, RADI returns 0.0. This possibility must be considered to avoid potential divide by zero errors.
RAGL(x)	Valid surface number	The global X-direction cosine of the ray following the surface. Valid only after a RAYTRACE or RAYTRACEX call.
RAGM(x)	Valid surface number	The global Y-direction cosine of the ray following the surface. Valid only after a RAYTRACE or RAYTRACEX call.
RAGN(x)	Valid surface number	The global Z-direction cosine of the ray following the surface. Valid only after a RAYTRACE or RAYTRACEX call.
RAGX(x)	Valid surface number	The global x coordinate of the ray intercept. Valid only after a RAYTRACE or RAYTRACEX call.
RAGY(x)	Valid surface number	The global y coordinate of the ray intercept. Valid only after a RAYTRACE or RAYTRACEX call.
RAGZ(x)	Valid surface number	The global z coordinate of the ray intercept. Valid only after a RAYTRACE or RAYTRACEX call.
RAND(x)	Positive numeric expression	Random floating point number uniformly distributed between 0 and the expression. It is recommend that the RANDOMIZE keyword be called before this function to change the random number seed. In particular, if NSC raytrace

		operands (NSTR, NSRA, etc.) are being used, RANDOMIZE is required to re-seed the random number generator.
RANX(x)	Valid surface number	The X-direction cosine of the surface normal. Valid only after a RAYTRACE or RAYTRACEX call.
RANY(x)	Valid surface number	The Y-direction cosine of the surface normal. Valid only after a RAYTRACE or RAYTRACEX call.
RANZ(x)	Valid surface number	The Z-direction cosine of the surface normal. Valid only after a RAYTRACE or RAYTRACEX call.
RAYE()	(null)	The ray-trace error flag, 0 if no error. Valid only after a RAYTRACE or RAYTRACEX call. See the RAYTRACE keyword for details.
RAYL(x)	Valid surface number	The X-direction cosine of the ray following the surface. Valid only after a RAYTRACE or RAYTRACEX call.
RAYM(x)	Valid surface number	The Y-direction cosine of the ray following the surface. Valid only after a RAYTRACE or RAYTRACEX call.
RAYN(x)	Valid surface number	The Z-direction cosine of the ray following the surface. Valid only after a RAYTRACE or RAYTRACEX call.
RAYO(x)	Valid surface number	The ray optical path length from the previous surface to the specified surface. The optical path length is the path length times the index of refraction, either or both of which may be negative. For rays inside a non-sequential surface, RAYO returns the sum of the path length times the index of refraction of all objects the ray passed through.Valid only after a RAYTRACE or RAYTRACEX call. See also OPTH and RAYT.
RAYT(x)	Valid surface number	The ray path length from the previous surface to the specified surface. The path length may be negative. Valid only after a RAYTRACE or RAYTRACEX call. See also OPTH and RAYO. For rays inside a non-sequential surface, RAYT returns the sum of the path lengths along all segments. See also RAYO.

RAYV()	(null)	0 if ray was not vignetted, else vignetted surface number. Valid only after a RAYTRACE or RAYTRACEX call.
RAYX(x)	Valid surface number	The X-coordinate of the ray intercept. Valid only after a RAYTRACE or RAYTRACEX call.
RAYY(x)	Valid surface number	The Y-coordinate of the ray intercept. Valid only after a RAYTRACE or RAYTRACEX call.
RAYZ(x)	Valid surface number	The Z-coordinate of the ray intercept. Valid only after a RAYTRACE or RAYTRACEX call.
RELI(f)	Relative illumination	RI for the specified field position.
SAGG(x,y,z)	x,y are the coordinates in lens units at the surface number z	Computes the sag in lens units at the specified point on the surface.
SCOM(A$, B$)	Any two string variable names	If the strings are equal, SCOM returns 0. If A$ is less than B$, then SCOM returns a value less than 0; otherwise, a value greater than 0.
SDIA(x)	Valid surface number	Semi-diameter of surface.
SIGN(x)	Numeric expression	Returns -1 if the argument is less than zero, 0 if the argument is zero, and +1 if the argument is positive.
SINE(x)	Numeric expression in radians	Sine of the expression.
SLEN(A$)	Any string variable name	The number of characters in the string variable A$
SOLM(operand, config, code)	Operand is the multi-configuration operand number. Config is the configuration number. Code is 0 for the status, 1 for the pickup configuration number, 2 for the pickup operand row, 3 for the pickup scale, 4 for the pickup offset, 5 for the current numerical or string value of the operand, 6 for the ZPL macro solve name, and 7 for the glass catalog name. Not all codes are used by all solve status types.	The return value is data about the solve type and data for the specified multi configuration operand and configuration. If the code is 0 for status, the returned value is 0 for fixed, 1 for variable, 2 for pick-up, 3 for thermal pick-up, 4 for ZPL Macro, and 5 for glass substitution solve. String values may be extracted using the $buffer() function after calling this function with a code that returns string data. For more information on the solve data and the meaning of the solve parameters, see "SOLVES".
SOLV(surf, code, param)	Surf is the surface number. Code is 0 for curvature, 1 for thickness, 2 for glass, 3 for conic, 4 for semidiameter, and 5 for TCE. For	The return value is data about the solve type for the specified surface and data. If param is zero, then an integer corresponding to the solve type is returned. For param between 1 and 3, the data is

	parameter data, the code is 100 plus the parameter number. For extra data, the code is 300 plus the extra data number. Param is an integer between 0 and 4, inclusive.	the solve parameters. For param 4, the data is the pickup column number. String values may be extracted using the $buffer() function after calling this function with a code that returns string data. For more information on the solve data and the meaning of the solve parameters, see "SOLVES"
SOSO(code)	Code is 0 to return the surface number of the macro solve currently executing, or 1 to return the object number.	This function returns the surface number and/or object number of the macro currently being executed as a solve. If the macro is not being used as a solve, these functions return meaningless data.
SPRO(surf, code)	Surf is the surface number. The code values are as defined for the SETSURFACEPROPERTY keyword. Only integer code values, and not mnemonics are supported.	Returns a numeric or string value from the property pages or editors of surfaces defined in the lens data editor. For information on the code values, see "SETSURFACEPROPERTY, SURP". String values may be extracted using the $buffer() function after calling this function with a code that returns string data. For example, to extract the comment column of surface 4 use these two commands: dummy = SPRO(4, 1) a$ = $buffer() The value dummy may be ignored. The string variable a$ will contain the surface comment. Properties where SETSURFACEPROPERTY requires the use of value2 are not supported by SPRO, use SPRX instead. These properties include parameter data, extra data, and coating layer multiplier and index offset values and status.
SPRX(surf, code, value2)	Surf is the surface number. The code and value2 values are as defined for the SETSURFACEPROPERTY keyword. Only integer code values, and not mnemonics are supported.	This function is similar to SPRO. The key difference is that SPRX supports one additional argument . Only parameters set by SETSURFACEPROPERTY that use the "value2" argument should use SPRX. See "SETSURFACEPROPERTY, SURP"
SQRT(x)	Positive numeric expression	Square root of the expression.
STDD() - obsolete	This function is obsolete, use SRPO instead.	
SURC(A$)	Any string variable name	Surface with comment. Returns the first surface number where the comment matches the string A$. The comparison is case insensitive. If no surface has the matching comment the function

		returns -1.
SVAL(A$)	Any string variable name	String value. Returns a floating point value of the string A$.
SYPR(code)	The code values are as defined for the SETSYSTEMPROPERTY keyword. Only integer code values, and not mnemonics are supported.	Returns a numeric or string value for the corresponding system data. For information on the code values , see "SETSYSTEMPROPERTY, SYSP". String values may be extracted using the $buffer() function after calling this function with a code that returns string data. For example, to extract the lens title use these two commands: dummy = SYPR(16) a$ = $buffer() The value dummy may be ignored. The string variable a$ will contain the lens title. Properties where SETSYSTEMPROPERTY requires the use of value2 are not supported by SYPR.
TANG(x)	Numeric expression in radians	Tangent of the expression.
Function	Argument	Return value
TMAS()	(null)	The total mass in grams of the lens from surface 1 to the image surface.
THIC(x)	Valid surface number	Thickness of the surface.
TOLV(op, col)	The integer op is the tolerance operand number, col is the column number in the Tolerance Data Editor. See discussion.	Returns a numeric or string value from the Tolerance Data Editor. If col is 1, 2, or 3, the return value is the integer 1, 2, or 3 value as used by that operand type. If col is 4 or 5, the min or max value of the tolerance is returned. If col is 6, the "Do Not Adjust During Inverse Tolerancing" flag is returned (1 for checked, 0 for unchecked). If col is 90, the nominal value of the tolerance is returned. If the col value is 0 or 99, the operand type name or operand comment is placed in the buffer string, respectively. For example, to extract the operand name of operand 7 use these two commands: dummy = TOLV(7, 0) a$ = $buffer() The value dummy may be ignored. The string variable a$ will contain the operand name. See

		also the string functions $TOLOPERAND and $TOLCOMMENT which perform a similar function.
UNIT()	(null)	Returns 0, 1, 2, or 3, if the current unit type is millimeters, centimeters, inches, or meters, respectively.
VEC1(x)	Positive subscript value	Returns the value of the array variable at the specified subscript.
VEC2(x)	Positive subscript value	Returns the value of the array variable at the specified subscript.
VEC3(x)	Positive subscript value	Returns the value of the array variable at the specified subscript.
VEC4(x)	Positive subscript value	Returns the value of the array variable at the specified subscript.
VERS()	(null)	Returns the version of the Zemax program. The version is of the form yymmdd. For example, the version for the June 15, 2005 release of Zemax will be 050615.
WAVL(x)	Valid wavelength number	Wavelength in micrometers.
WINL()	(null)	Returns the number of the analysis window most recent ly opened by the keyword OPENANALYSISWINDOW.
WINN()	(null)	Returns the number of open analysis windows.
WWGT(x)	Valid wavelength number	Wavelength weighting.
XMIN()	(null)	The minimum x-coordinate in the graphics window.
XMAX()	(null)	The maximum x-coordinate in the graphics window.
Function	Argument	Return value
YMIN()	(null)	The minimum y-coordinate in the graphics window.
YMAX()	(null)	The maximum y-coordinate in the graphics window.

Using the FICL() function

The FICL function computes fiber coupling. Because the function takes so many arguments, the arguments are placed in a vector array and then the function is called. The vector array must be one of the four VEC arrays already defined. The values placed in the vector array are defined as follows:

```
0 = Sampling (1 for 32 x 32, 2 for 64 x 64, 3 for 128 x 128, etc..)
1 = Wavelength #
2 = Field #
3 = Ignore source flag (0 for false, 1 for true)
4 = Source NA x-direction
5 = Receiver NA x-direction
6 = Source x angle (deg)
7 = Source y angle (deg)
8 = Receiver tilt about x (deg
9 = Receiver tilt about y (deg)
10 = Receiver decenter x
11 = Receiver decenter y
12 = Receiver decenter z
13 = Align source to chief ray flag (0 for false, 1 for true)
14 = Align receiver to chief ray flag (0 for false, 1 for true)
15 = Use polarization flag (0 for false, 1 for true)
16 = Source NA y-direction (if zero then NAy = NAx)
17 = Receiver NA y-direction (if zero then NAy = NAx)
18 = Use Huygens integral (0 for false, 1 for true)
```

The fiber coupling is computed by calling FICL(n) where n is the vector number containing the argument list.

String functions

The following table lists the available string functions.

ZPL STRING FUNCTIONS

Function	Description
$BUFFER()	Returns the current string in the lens buffer. This function is used to extract string data from various ZPL keywords and functions.
$CALLSTR(i)	Returns the string from the CALLMACRO string buffer at index i. See "Calling a Macro from within a Macro".
$COAT(i)	Returns the coating name for the ith surface.
$COATINGPATH()	Returns the path name for coating files.
$COMMENT(i)	Returns the comment string for the ith surface.
$DATAPATH()	Returns the path name for data files.
$DATE()	Returns the current date and time string. The formatting is specified by the Date/ Time control settings described under "Miscellaneous".
$EXTENSIONPATH()	Returns the path name for Zemax extensions.

$FILENAME()	Returns the current lens file name, without the path.
$FILEPATH()	Returns the current lens file name, with the complete path.
$GETSTRING(A$, n)	Returns the nth sub-string for the string A$ using spaces for delimiters. For example, if A$ = "one two three", then $GETSTRING(A$, 2) returns "two".
$GETSTRINGC(A$, n)	Returns the nth sub-string for the string A$ using commas for delimiters. For example, if A$ = "one,two,three", then $GETSTRING(A$, 2) returns "two".
$GLASS(i)	Returns the glass name of surface number i.
$GLASSCATALOG(i)	Returns the name of the ith loaded glass catalog for the current lens. If i is less than 1, then the names of all the loaded catalogs separated by spaces are returned in a single string.
$GLASSPATH()	Returns the path name for glass catalog files.
$LEFTSTRING(A$, n)	Returns the left most n characters in the string A$. If A$ has fewer than n characters, the remaining spaces will be padded with blanks. This allows formatting of strings with a fixed length.
$LENSNAME()	Returns the lens title defined in the System Explorer.
$MACROPATH()	Returns the path name for macro files.
$NOTE(line#)	Returns the notes information defined in the System Explorer. Because the notes may be very long, $NOTE returns the characters from the notes in groups called lines. A line ends when a newline (carriage return) character is found, or when the total number of consecutive characters on the line reaches 100, whichever comes first. The line# indicates which line in the notes is to be returned.

For example, $NOTE(1) returns the first line in the notes, up to the first newline character. If there are more than 100 consecutive characters before the first newline is found, then line 1 will be the first 100 characters, and line 2 will be the remainder of the line up to the first newline, or the next 100 characters, whichever comes first.

The maximum number of characters in the notes is currently 4000 (because of space limitations, it may not be possible to edit this many characters in the Title/Notes section of the System Explorer however). $NOTE will return a null (empty) string if there are no defined characters in the notes for the specified line.

The function SLEN can be used to determine the actual number of characters returned by $NOTE using this syntax: |

	A$ = $NOTE(1) N = SLEN(A$)
	The value N will be the integer number of characters in the string A$.
$OBJECTPATH()	Returns the path name for NSC object files.
$PATHNAME()	Returns the path name only for the current lens file. This is useful for determining the folder where the lens file is stored.
$PROGRAMPATH()	Returns the path name for program files.
$QUOTE()	Returns the double quote character (").
$RIGHTSTRING(A$, n)	Returns the right most n characters in the string A$. If A$ has fewer than n characters, the remaining spaces will be padded with blanks. This allows formatting of strings with a fixed length.
$STR(expression)	Returns a string formatted using the format defined by the FORMAT keyword. The numeric expression may be any equation, including combinations of constants, variables, and functions. See function SVAL(A$) to convert strings to numbers.
$TAB()	Returns the tab character (\t).
$TEMPFILENAME()	Returns the name of a temporary file, with complete path, suitable for temporary storage of text or binary data. See keyword GETTEXTFILE.
$TOLCOMMENT(operand)	Returns the comment for the specified tolerance operand.
$TOLOPERAND(operand)	Returns the operand name for the specified tolerance operand.
$UNITS()	Returns either MM, CM, IN, or M, depending upon the current lens units.

Keywords

Keywords provide the capability to direct program flow, to generate output, and to perform certain crucial tasks, such as ray tracing and modifying the lens prescription. Each keyword is described in detail in the following sections.

APMN, APMX, APTP, APXD, APYD

These commands are obsolete. See "SETSURFACEPROPERTY, SURP" .

ATYP, AVAL

These commands are obsolete. See "SETSYSTEMPROPERTY, SYSP".

BEEP

Makes an audible beep sound.
Syntax:

```
BEEP
```

Discussion:
This command can be used to alert the user when a calculation is finished or input is required.

CALLMACRO
Calls another ZPL macro.
Syntax:
```
CALLMACRO filename.zpl
CALLMACRO name$
```

Discussion:
This command is used to call another ZPL macro. The filename should be the full macro name without the path. The file must exist in the folder for ZPL macro files. When the macro returns, any text output generated by the called macro will be copied into the calling macro's text output window. See "Calling a Macro from within a Macro".

CALLSETDBL
Sets a double precision numeric value in the master macro's buffer.
Syntax:
```
CALLSETDBL index, value
```

Discussion:
This command is used to set a numeric value into the buffer maintained by the master macro. The index can be any value between 0 and 50, inclusive. The numeric value may be retrieved using the CALD function. See "Calling a Macro from within a Macro".

CALLSETSTR
Sets a string value in the master macro's buffer.
Syntax:
```
CALLSETSTR index, text$
```

Discussion:
This command is used to set a string value into the buffer maintained by the master macro. The index can be any value between 0 and 50, inclusive. The string value may then be retrieved using the $CALLSTRING function. See "Calling a Macro from within a Macro".

COAT
This command is obsolete. See "SETSURFACEPROPERTY, SURP".

CLOSE
Closes a file previously opened by the OPEN command.
Syntax:
```
CLOSE
```

Discussion:
See the description for OPEN.

CLOSEWINDOW
Suppresses the display of the default output window. This keyword can also be used to close any open analysis windows.
Syntax:
```
CLOSEWINDOW
CLOSEWINDOW n
```

Discussion:

CLOSEWINDOW (with no argument "n" provided) is used to run the ZPL macro in "quiet" mode. The text window normally displayed at the end of the macro execution will not be displayed if the CLOSEWINDOW keyword is included at any line in the macro. CLOSEWINDOW has no other effect on macro execution.
CLOSEWINDOW (with an integer argument "n" provided) will close analysis window number n.

COLOR
Sets the current pen color to use for graphics line and text functions.
Syntax:
```
COLOR n
```

Discussion:
The value n must evaluate to an integer corresponding to the desired color. The new pen color will be used for all subsequent line and text commands in graphics mode. Black is color 0, the other colors are as defined in "Colors 1-12, Colors 13-24". There are 24 colors other than black available.

COMMAND
Executes a shell command.
Syntax:
```
COMMAND executable, arguments
```

Discussion:
The values executable and arguments are either string literals or string variables. The value of executable is the name (including folder path) to the executable to run. The value of arguments is optional, and includes any command line arguments to the executable.
Once the command is launched, the executable runs in its own thread. This command does not cause the ZPL macro or Zemax to "wait" until the executable completes its task. No error codes are returned, and Zemax has no way of knowing if the shell command was launched successfully.

COMMENT
This command is obsolete. See "SETSURFACEPROPERTY, SURP".

CONI
This command is obsolete. See "SETSURFACEPROPERTY, SURP".

CONVERTFILEFORMAT
Converts a text file from ANSI to Unicode format or from Unicode to ANSI format.
Syntax:
```
CONVERTFILEFORMAT filename, encoding
```

Discussion:
Filename is the name of the text file to convert. Encoding is 1 to convert the file from Unicode to ANSI, or 2 to convert ANSI to Unicode. Note converting a Unicode file into ANSI may cause the loss of data, if any of the characters have no ANSI equivalent.

COPYFILE
COPYFILE is used to make a copy of a file.
Syntax:
```
COPYFILE sourcefilename, newfilename
```

Discussion:
This keyword requires two file names, defined as literal string expressions in quotes or as string variables. The file sourcefilename is copied into the new file newfilename. If newfilename already exists it is overwritten without warning.
Example:
```
COPYFILE "C:\source.dat", "C:\copy.dat"
```

Related Keywords:

DELETEFILE
RENAMEFILE

CURV

This command is obsolete. See "SETSURFACEPROPERTY, SURP".

DECLARE

See "Array variables".

DEFAULTMERIT

Generates a default merit function.

Syntax:
```
DEFAULTMERIT type, data, reference, method, rings, arms, grid, delete, axial, lateral,
start, xweight, oweight, pup_obsc
```

Discussion:

This keyword generates a default merit function in the Merit Function Editor. Any existing default merit function will be deleted. For details see "Modifying the merit function". The values are as follows:

type: use 0 for RMS, 1, for PTV.

data: use 0 for wavefront, 1 for spot radius, 2 for spot x, 3 for spot y, 4 for spot x + y. reference: use 0 for centroid, 1 for chief, 2 for unreferenced.

method: use 1 for Gaussian quadrature, 2 for rectangular array. rings: the number of annular rings (Gaussian quadrature only).

arms: the number of radial arms (Gaussian quadrature only). The number of arms must be even and no less than 6.

grid: the size of the grid. Use an integer, such as 8, for an 8 x 8 grid. n must be even and no less than 4. delete: use 0 to not delete vignetted rays, 1 to delete vignetted rays.

axial: use -1 for automatic, which will use symmetry only if the system is axial symmetric. Use 1 to assume axial symmetry, 0 to not assume axial symmetry.

lateral: use 1 to ignore lateral color, 0 otherwise.

start: use -1 for automatic, which will add the default merit function after any existing DMFS operand. Otherwise use the operand number at which to add the default merit function. Any existing operands above the specified operand number will be retained.

xweight, oweight: the x direction weigh and overall weight for the merit function. Only the data "spot x + y" uses the xweight value.

pup_obsc: the pupil obscuration ratio.

DELETE

Deletes a surface from the spreadsheet.

Syntax:
```
DELETE n
```

Discussion:

The value n must evaluate to an integer expression. See also INSERT.

Example:
```
DELETE 5
DELETE i+2*j
```

DELETECONFIG

DELETECONFIG deletes an existing configuration in the multi-configuration editor.

Syntax:
```
DELETECONFIG config
```

Discussion:

The value config must evaluate to an integer expression greater than 0 and less than or equal to the current number of configurations. See also INSERTCONFIG and DELETEMCO.

DELETEFILE

DELETEFILE is used to delete a file.

Syntax:
```
DELETEFILE filename
```

Discussion:
This keyword requires a file names, defined as literal string expression in quotes or as a string variable.

Example:
```
DELETEFILE XFILE$
```

Related

Keywords

:

COPYFILE RENAMEFILE

DELETEMCO

DELETEMCO deletes an existing operand in the multi-configuration editor.

Syntax:
```
DELETEMCO row
```

Discussion:
The value row must evaluate to an integer expression greater than 0 and less than or equal to the current number of operands. See also INSERTMCO and DELETECONFIG.

DELETEMFO

DELETEMFO deletes an existing operand in the merit function editor.

Syntax:
```
DELETEMFO row
DELETEMFO ALL
```

Discussion:
The value row must evaluate to an integer expression greater than 0 and less than or equal to the current number of operands. If the ALL argument is used all operands are deleted. See also INSERTMFO.

DELETEOBJECT

Deletes an existing NSC object.

Syntax:
```
DELETEOBJECT surf, object
```

Discussion:
The value surf must evaluate to an integer expression, and the specified surface must be a non-sequential component surface. Use a surf value of 1 if the program mode is non-sequential. The value object must evaluate to an integer that is between 1 and the current number of objects plus 1, inclusive. The specified object will be deleted, renumbering other objects as required. See also INSERTOBJECT and SETNSCPROPERTY.

Example:
```
DELETEOBJECT 1, 1
```

DELETETOL

DELETETOL deletes an existing operand in the tolerance data editor.

Syntax:
```
DELETETOL row
```

Discussion:
The value row must evaluate to an integer expression greater than 0 and less than or equal to the current number of operands. See also INSERTTOL and SETTOL.

EDVA

This command is obsolete. See "SETSURFACEPROPERTY, SURP".

END

See GOSUB.

EXPORTBMP

Exports any graphic window as a BMP file.

Syntax:
```
EXPORTBMP winnum, filename, delay
```

Discussion:

The winnum value may be either an integer or an expression that evaluates to an integer. The integer winnum corresponds to the <u>graphic</u> window number that should be saved to a file. Zemax numbers windows sequentially as they are opened, starting with 1. Any closed windows are deleted from the window list, without renumbering the windows which remain. Any windows opened after another window has been closed will use the lowest window number available. The filename should be the full file name including the path, but with no extension. Zemax will automatically add the BMP extension. The optional delay parameter specifies a time delay in milliseconds. For some complex graphics, a delay is required to allowed the graphic to be completely redrawn and the screen capture to complete. If the BMP files appear incomplete, try a delay value of 500 - 2500 milliseconds. See also EXPORTJPG.

Example:
```
EXPORTBMP 1, "C:\TEMP\MYBMPFILE"
EXPORTBMP k, A$
```

EXPORTCAD

Exports lens data as an IGES, STEP, SAT, or STL file for import into CAD programs.

Syntax:
```
EXPORTCAD filename
```

Discussion:

For a full description of CAD file export, see "Export CAD File". The filename can be a literal string, such as "C:\DATA\FILE.IGS" or a string variable name, such as MYFILENAME$. There are many other parameters required to define the export. These other data are placed in the vector 1 variable as follows:

VEC1(1): The file type. Use 0 for IGES, 1 for STEP, 2 for SAT, 3 for STL.

VEC1(2): The number of spline points to use (if required on certain entity types). Use 16, 32, 64, 128, 256, or 512.

VEC1(3): The First surface to export. In NSC Mode, this is the first object to export. VEC1(4): The last surface to export. In NSC Mode, this is the last object to export. VEC1(5): The layer to place ray data on.

VEC1(6): The layer to place lens data on.

VEC1(7): Use 1 to export dummy surfaces, otherwise use 0. VEC1(8): Use 1 to export surfaces as solids, otherwise use 0.

VEC1(9): Ray pattern. Use 0 for XY, 1 for X, 2 for Y, 3 for ring, 4 for list, 5 for none, 6 for grid, and 7 for solid beams.

VEC1(10): The number of rays.

VEC1(11): The wave number. Use 0 for all. VEC1(12): The field number. Use 0 for all.

VEC1(13): Use 1 to delete vignetted rays, otherwise use 0. VEC1(14): The dummy surface thickness in lens units.

VEC1(15): Use 1 to split rays from NSC sources, otherwise use 0. VEC1(16): Use 1 to scatter rays from NSC sources, otherwise use 0.

VEC1(17): Use 1 to use polarization when tracing NSC rays, otherwise use 0. Polarization is automatically selected if splitting is specified.

VEC1(18): Use 0 for the current configuration, 1 ton for a specific configuration where n is the total number of configurations, n+1 to export "All By File", n+2 to export "All By Layer", and n+3 for "All At Once". For a more detailed explanation of the configuration setting, see "Export CAD File".

VEC1(19): Tolerance setting. Use 0 for 1.0E-4, 2 for 1.0E-05, 3 for 1.0E-06, and 4 for 1.0E-07.

VEC1(20), VEC1(21): If the program mode is sequential, and the range of surfaces includes a non-sequential components surface, these values allow a range of objects to be exported. VEC1(20) is the first object to export, and VEC1(21) is the last object to export. If both values are zero or out of range all objects are exported.

Example:
```
VEC1(1) = 0
VEC1(2) = 32
VEC1(3) = 1
etc...
VEC1(18) = 0
EXPORTCAD "C:\TEMP\MYCADFILE.IGS"
```

EXPORTJPG
Exports any graphic window as a JPG file.
Syntax:
```
EXPORTJPG winnum, filename, delay
```

Discussion:
See EXPORTBMP.
Example:
```
EXPORTJPG 3, "C:\TEMP\MYJPGFILE"
EXPORTJPG k, FILENAME$, 500
```

FINDFILE
Used to find the names of files.
Syntax:
```
FINDFILE TEMPNAME$, FILTER$
```

Discussion:
This keyword requires two expressions, one to specify the string variable name to store the file name in, and another string variable which contains a "filter" string. The filter string usually specifies a path name and wildcards appropriate to the desired file type. See the example below.
FINDFILE is useful for listing all files of a certain type in a folder, or for analyzing large numbers of similar lens files. To reset FINDFILE back to the first file of any type, just call FINDFILE with a different filter, then call FINDFILE again with the original filter name. Each time FINDFILE is called with a new filter, it resets back to the first file that meets the filter specifications.
Example:
```
FILTER$ = "C:\Zemax\*.ZMX"
PRINT "Listing of all Zemax files in ", FILTER$
FINDFILE TEMPFILE$, FILTER$
LABEL 1
if (SLEN(TEMPFILE$))
    PRINT TEMPFILE$
    FINDFILE TEMPFILE$, FILTER$
    GOTO 1
ENDIF
PRINT "No more files."
```

FLDX, FLDY, FWGT, FVDX, FVDY, FVCX, FVCY, FVAN
These commands are obsolete. See "SETSYSTEMPROPERTY, SYSP".

FOR, NEXT
The FOR and NEXT keywords define a program block which is executed a specific number of times in a loop.
Syntax:
```
FOR variable, start_value, stop_value, increment commands...
NEXT
```

Discussion:

The keyword FOR marks the beginning of a group of commands to be executed a multiple number of times. FOR requires a variable to be specified which acts as a counter (it need not be an integer), a starting value for the counter, a stop value, and an increment. The NEXT keyword marks the end of the group of commands. FOR- NEXT loops may be nested. The number of FOR and NEXT commands must be the same.

Upon reaching a FOR command, the expressions for the start, stop, and increment values are evaluated and saved. The stop and increment values are not evaluated again, even if the expressions defining the values consist of variables whose values change within the program block. Only the values valid at the beginning of the FOR loop are used.

If the start value and stop value are the same, the loop executes exactly once. If the start value is less than the stop value, then the loop continues until the counter variable is greater than the stop value. If the start value is greater than stop value, then the loop continues until the counter variable is less than the stop value.

The increment value should always be an integer. See the first example below for a simple way to loop through non-integer values.

Example:
```
FOR i, 1, 10, 1
     a = 0.2*i
     PRINT a
NEXT


j = 5 k = 0
FOR i, j, j + 5, 2
     k = i + j + k
NEXT
```

FORMAT

Specifies the numerical precision format for subsequent PRINT and $STR commands.

Syntax:
```
FORMAT m.n
FORMAT m.n EXP
FORMAT m [INT]
FORMAT "C_format_string" [LIT]
```

Discussion:
The integers m and n are separated by a decimal point. The values for m and n used must be explicit, i.e. values stored as variables cannot be used. The value m refers to the total number of characters to be printed, even though some of them may be blank. The value n refers to the number of places to display after the decimal point. Therefore, FORMAT 8.4 will cause subsequent PRINT commands to print 8 characters, with 4 numbers after the decimal point. FORMAT .5 will cause PRINT to show 5 decimal places, and as many total places as needed. FORMAT only affects numeric output from PRINT. If a number is too large to fit within the m decimal places, then the m portion of the FORMAT command will be ignored.

The optional keyword EXP after the m.n expression indicates exponential notation should be used.

The optional keyword INT indicates the value should be first converted to an integer and printed in integer format using the number of places specified by m.

The optional keyword LIT (for literal) indicates the value should be printed according to the "C" language format specifier. The C format specification can be found in any programming reference for the C language.

Example:
The macro:
```
X = 5
FORMAT 5.3
PRINT "FORMAT 5.3         :", X
FORMAT 12.2 EXP
PRINT "FORMAT 12.2 EXP   :", X
FORMAT 6 INT
PRINT "FORMAT 6 INT      :", X
FORMAT "%#06i" LIT
PRINT "FORMAT %#06i LIT  :", X
```

will produce this output:

```
FORMAT 5.3        :5.000
FORMAT 12.2 EXP   :  5.00E+000
FORMAT 6 INT      :    5
FORMAT %#06i LIT  :000005
```

FTYP

This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

GCRS

This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

GDATE

GDATE will place the current date under the lens name in the text box on user defined graphics screens.
Syntax:
```
GDATE
```

Discussion:
GDATE is primarily used for making graphics generated by ZPL macros look like other Zemax graphics.
Example:
See "GRAPHICS".

GETEXTRADATA

Retrieves the extra data values loaded into the lens data editor. The data is placed in one of the vector array variables
(either VEC1, VEC2, VEC3, or VEC4).
Syntax:
```
GETEXTRADATA vector_expression, surface_expression
```

Discussion:
The data is stored in the specified VECn array variable. For example, if the command GETEXTRADATA 1, 5 is issued,
the extra data for surface 5 will be placed in VEC1. The data is stored in the following format, where the first number in
each line refers to the array position:
```
0:    The number of extra data values in the vector
1:    The first extra data value
n:    The nth extra data value
```

See "Extra data" for descriptions of the extra data values.

GETGLASSDATA

Retrieves the data for any glass in the current catalogs. The data is placed in one of the vector array variables (either VEC1,
VEC2, VEC3, or VEC4).
Syntax:
```
GETGLASSDATA vector_expression, glass_number
```

Discussion:
The data is stored in the specified VECn array variable. For example, if the command GETGLASSDATA 1, 32 is issued,
the data for glass number 32 will be placed in VEC1. The glass number is returned by the GNUM function. The data is
stored in the following format, where the first number in each line refers to the array position:
```
0:    The number of data values in the vector
1:    Formula number: The number indicates the formula as follows:
      1 = Schott, 2 = Sellmeier 1, 3 = Herzberger, 4 = Sellmeier 2
      5 = Conrady,6 = Sellmeier 3, 7 = Handbook of Optics 1, 8 = Handbook of Optics 2
      9 = Sellmeier 4,10 = Extended, 11 = Sellmeier 5, 12 = Extended 2
2:    Reference temperature in degrees c
3:    Nd
4:    Vd
5:    Thermal coefficient of expansion -30 to +70 c
6:    Thermal coefficient of expansion +20 to 300 c
```

```
7:    Density in g/cm^3
8:    Deviation from normal line P gf
9:    Lambda min
10:   Lambda max
11-16:Constants of dispersion A0-A5 (meaning depends upon formula)
17-22:Thermal constants of dispersion
23-(22 + #waves): Internal transmission coefficient (per mm) alpha, T = exp(alpha *
path). The alpha for wavelength 1 is stored in 23, wavelength 2 is in 24, etc., up to the
number of wavelengths used by the system.
(23 + #waves) - (32 + #waves): Constants of dispersion A0-A9 (meaning depends upon
formula)
```

Related Functions:
GNUM

GETLSF

Calculates the geometric edge and line response functions, similar to the "Geometric Line/Edge Spread".
Syntax:
```
GETLSF wave, field, sampling, vector, maxradius, use_polarization
```

Discussion:
Wave is an integer corresponding to the wavelength number to use for the calculation. A value of zero indicates a polychromatic calculation. Field must be an integer between 1 and the maximum number of fields. The value indicates which field position to use. Sampling may be 1 (32 x 32), 2 (64 x 64), 3 (128 x 128), etc... up to 2048 x 2048. The vector argument must be an integer value between 1 and 4, and specifies which vector array the data should be placed in. The maxradius argument is the maximum radial coordinate of the edge and line spread functions; this is the half-width of the data range. Use 0 for a default width. If any of the arguments fall outside the valid ranges, then the nearest acceptable value is used instead.
The data is returned as an array of values in the specified vector. Vector position 0-3 will hold the number of points "N", the starting x coordinate (this is the negative of the half width of the data range), the delta coordinate, and the offset (defined below), respectively. The offset is the first position in the vector that holds the edge or line spread data. Starting at the offset, the first N value are the tangential LSF response. The next N values are the sagittal LSF response. The tangential and sagittal ERF values are in the next two groups of N data values.
If the current vector size is not large enough, Zemax will automatically increase the size of the vectors to hold the LSF data in the manner described in SETVECSIZE.
Example:
```
! Macro computes and prints the LSF and ERF for polychromatic light at field 1.
!
! Syntax is GETLSF wave, field, samp, vector, maxradius, usepol
!
GETLSF 0, 1, 3, 1, 0, 0
N_BINS = vec1(0)
STARTX = vec1(1)
DELTAX = vec1(2)
OFFSET = vec1(3)
FORMAT 15.0
PRINT "Number of Bins    = ", N_BINS
FORMAT 15.3 EXP
PRINT "Starting Coordinate = ", STARTX
PRINT "Delta Coordinate  = ", DELTAX
FORMAT 15.0
PRINT "Offset            = ", OFFSET
OFF1 = OFFSET
OFF2 = OFF1 + N_BINS
OFF3 = OFF2 + N_BINS
OFF4 = OFF3 + N_BINS
MAXI = N_BINS-1
FORMAT 16.3 EXP
PRINT
PRINT "            X           TLSF        SLSF        TERF        SERF"
```

```
 PRINT
 FOR i, 0, MAXI, 1
       PRINT STARTX + DELTAX*i,
       PRINT vec1(OFF1 + i),
       PRINT vec1(OFF2 + i),
       PRINT vec1(OFF3 + i),
       PRINT vec1(OFF4 + i)
 NEXT i
```

GETMTF

Calculates tangential and sagittal MTF, real part, imaginary part, phase, or square wave response data for the currently loaded lens file, and places the data in one of the vector arrays (either VEC1, VEC2, VEC3, or VEC4).

Syntax:
```
GETMTF freq, wave, field, sampling, vector, type
```

Discussion:
The freq argument is the desired spatial frequency in MTF Units (see "MTF Units"). If the frequency is less than zero, or greater than the cutoff frequency, GETMTF returns zero. Wave is an integer corresponding to the wavelength number to use for the calculation. A value of zero indicates a polychromatic calculation. Field must be an integer between 1 and the maximum number of fields. The value indicates which field position to use. Sampling may be 1 (32 x 32), 2 (64 x 64), 3 (128 x 128), etc... up to 2048 x 2048. The vector argument must be an integer value between 1 and 4, and specifies which vector array the data should be placed in. The type argument refers to the data type: 1 for MTF, 2 for real part, 3 for imaginary part, 4 for phase in radians, 5 for square wave MTF. If any of the arguments fall outside the valid ranges, then the nearest acceptable value is used instead. This calculation uses the FFT MTF method (see "FFT MTF").
The data is returned in one of the vector arrays with the following format: Vector position 0: tangential response; Vector position 1: sagittal response.
Example:
```
! This macro computes the T & S response at 30 lp/mm
! for the currently loaded lens, polychromatic,
! at the maximum defined field,
! and a 32x32 grid density (sampling = 1).
! Data will be placed in vector 1.
! This is all it takes to get the data:
GETMTF 30, 0, NFLD(), 1, 1, 1
PRINT "Tangential response:", vec1(0)
PRINT "Sagittal response :", vec1(1)
```

GETPSF

Calculates the diffraction point spread function (PSF) using the FFT algorithm and places the data in one of the vector arrays (either VEC1, VEC2, VEC3, or VEC4).
Syntax:
```
GETPSF wave, field, sampling, vector, unnormalized, phaseflag, imagedelta
```

Discussion:
Wave is an integer corresponding to the wavelength number to use for the calculation. A value of zero indicates a polychromatic calculation. Field must be an integer between 1 and the maximum number of fields. The value indicates which field position to use. Sampling may be 1 (32 x 32), 2 (64 x 64), 3 (128 x 128), etc... up to 2048 x 2048. The vector argument must be an integer value between 1 and 4, and specifies which vector array the data should be placed in. The unnormalized flag is zero if the data should be normalized to a peak of 1.0, if the unnormalized value is 1, then the data is returned unnormalized. If phase flag is zero, the data returned is intensity, if 1, then the phase in degrees is returned. The imagedelta value is the spacing between PSF points in micrometers; use zero for the default spacing. The wavelength must be monochromatic to compute phase data. If any of the arguments fall outside the valid ranges, then the nearest acceptable value is used instead.
The data is returned in one of the vector arrays with the following format:
Vector position 0: the total number of PSF data points in the vector array. Usually, this number will be 4*n*n where n is the sampling size (32, 64, etc.). For example, if the sampling density is 2, the pupil sampling will be 64 x 64, and there will be 128 x 128 or 16,384 values in the array. This will require 8 bytes per number, or a total of 131 kb. A sampling density of 1024 will require at least 8 Mb just for the array; another 64 Mb or more to compute the PSF. Position 0 also returns other values as error codes. If position 0 is zero, then the computation was aborted. If -1, then the vector array is not large enough to hold all the data. Use SETVECSIZE to make the array bigger. If -2, then there is not enough system RAM to compute the PSF data. If -3, a general error occurred while computing the PSF.
Vector position 1 through 4*n*n holds the PSF data intensity. The first 2n values are the first row, going left to right from -x to +x, then each subsequent block of 2n values is another row, going from -y to +y. Vector position 4*n*n+1 holds the spacing between data values in micrometers.
Example:
```
! This macro computes the PSF
! for the currently loaded lens, polychromatic,
! at the first field,
! and a 32x32 grid density (sampling = 1),
```

```
! data will be placed in vector 1,
! normalized to 1,
! no phase data,
! default image delta.

SETVECSIZE 4500
GETPSF 0, 1, 1, 1, 0, 0, 0
np = vec1(0)
IF (np == 0)
    PRINT "PSF Computation aborted."
    GOTO 1
ENDIF
IF (np == -1)
    PRINT "SETVECSIZE too small for PSF data."
    GOTO 1
ENDIF
IF (np == -2)
    PRINT "Not enough system RAM for PSF data."
    GOTO 1
ENDIF
PRINT "There are ", np, " data points, spaced ", vec1(np+1), " micrometers apart".
LABEL 1
```

GETSYSTEMDATA

Retrieves most system specific data, such as effective focal length, working F/#, apodization factors, and other data not associated with any particular surface. The data is placed in one of the vector array variables (either VEC1, VEC2, VEC3, or VEC4).

Syntax:
```
GETSYSTEMDATA vector_expression
```

Discussion:
The data is stored in the specified VECn array variable. For example, if the command GETSYSTEMDATA 1 is issued, the system data will be placed in VEC1. See also SETSYSTEMPROPERTY.

The data is stored in the following format, where the first number in each line refers to the array position:
```
 0:    The number of system data values in the vector
 1:    Aperture Value
 2:    Apodization Factor
 3:    Apodization Type (0:none, 1:gaussian, 2:tangent)
 4:    Adjust Index Data To Environment setting (1 if true, 0 if false)
 5:    Temperature in degrees c (valid only if Use Env Data true)
 6:    Pressure in ATM (valid only if Use Env Data true)
 7:    Effective Focal Length
 8:    Image Space F/#
 9:    Object Space Numerical Aperture
10:    Working F/#
11:    Entrance Pupil Diameter
12:    Entrance Pupil Position
13:    Exit Pupil Diameter
14:    Exit Pupil Position
15:    Paraxial Image Height
16:    Paraxial Magnification
17:    Angular Magnification
18:    Total Track
19:    Ray Aiming (0 for off, 1 for paraxial, and 2 for real)
20:    X Pupil Shift
21:    Y Pupil Shift
22:    Z Pupil Shift
23:    Stop Surface Number
24:    Global Coordinate Reference Surface Number
25:    Telecentric object space (0 for off, 1 for on)
26:    The number of configurations
```

```
27:    The number of multi-configuration operands
28:    The number of merit function operands
29:    The number of tolerance operands
30:    Afocal image space (0 for off, 1 for on)
31:    X Pupil Compress
32:    Y Pupil Compress
```

GETTEXTFILE

Creates a text file from any Zemax analysis window that supports text.
Syntax:
```
GETTEXTFILE textfilename, type, settingsfilename, flag
```

Discussion:
The textfilename must be in quotes, or be a string variable name, and include the full path, name, and extension for the file to be created. The string function $TEMPFILENAME can be used to define a suitable temporary file name.
The type argument is a 3 character string code that indicates the type of analysis to be performed. For the full list of string codes, see the "String Codes" section of the The Programming Tab. If no type is provided or recognized, a standard ray trace will be generated.
If a valid file name in quotes or a string variable name is used for the settingsfilename argument, Zemax will use or save the settings used to compute the text file, depending upon the value of the flag parameter.
If the flag value is 0, then the default settings will be used. If the lens file has it's own default settings, then those will be used; these are the settings stored in the "lensfilename.cfg" file. If no lens specific default settings exist, then the default settings for all Zemax files, stored in the file "Zemax.CFG" will be used, if any. If no previous settings have been saved for this or any other lens, then the default settings used are the "factory" defaults used by Zemax.
If the flag value is 1, then the settings provided in the settings file, if valid, will be used to generate the file. If the data in the settings file is in anyway invalid, then the default settings will be used to generate the file. The only valid settings files are those generated by Zemax, then renamed and saved to a new user defined file name. For example, on the Spot Diagram settings dialog, pressing "Save" will generate a "SPT.CFG" file with the saved settings. If this file is renamed to "MySPT.CFG", then the file, with a full path, may be used as the settingsfilename argument to GetTextFile.
If the flag value is 2, then the settings provided in the settings file, if valid, will be used and the settings box for the requested feature will be displayed. After the user makes any changes to the settings the file will then be generated using the new settings. The dialog boxes used to change the analysis settings use the data from the lens currently in the Lens Data Editor.
No matter what the flag value is, if a valid file name is provided for the settingsfilename, the settings used will be written to the settings file, overwriting any data in the file. To modify the settings defined within an existing settings file, use MODIFYSETTINGS.
Only text, and not graphic files, are supported by GETTEXTFILE.
See also OPEN, CLOSE, READ, READNEXT, READSTRING, and MODIFYSETTINGS.
Example:
```
! Macro sample to generate cardinal points in a file.
! Get a temporary file name
A$ = $TEMPFILENAME()
! Compute the data and place in the temp file
GETTEXTFILE A$, Car
! Open the new file and print it out
OPEN A$
LABEL 1
READSTRING B$
IF (EOFF()) THEN GOTO 2
PRINT B$
GOTO 1
LABEL 2
CLOSE
```

GETVARDATA

Retrieves the current number, type, and value of all optimization variables. The data is placed in one of the vector array variables (either VEC1, VEC2, VEC3, or VEC4).

Syntax:
```
GETVARDATA vector
```

Discussion:
The data is stored in the specified VECn array variable. For example, if the command GETVARDATA 1 is issued, the data will be stored in VEC1. The data is stored in the following format, where the first number in each line refers to the array position:
```
0:      n, the number of variables
1:      The type code for the first variable
2:      Surface number for the first variable
3:      Parameter number for the first variable
4:      Object number for the first variable
5:      The value of the first variable
5*q-4: The type code for the qth variable
5*q-3: Surface number for the qth variable
5*q-2: Parameter number for the qth variable
5*q-1: Object number for the qth variable
5*q:   The value of the qth variable
etc...
```

The integer q goes from 1 to n, where n is the number of variables. If n is zero, then no valid data is returned. The value of the number in array position zero, n, is always valid. The type codes for variables is as described in the following table. The surface number, parameter number, and object number may or may not have meaning depending upon the type of variable.

GETVARDATA TYPE AND ID CODES

Variable type	Type Code	Surface	Parameter	Object
Curvature	1	surface #	-	-
Thickness	2	surface #	-	-
Conic	3	surface #	-	-
Index Nd	4	surface #	-	-
Abbe Vd	5	surface #	-	-
Partial Dispersion $\Delta P_{g,F}$	6	surface #	-	-
TCE	7	surface #	-	-
Parameter Values	8	surface #	parameter #	-
Extra Data Values	9	surface #	extra data #	-
Multi-configuration Operand Values	10	oper #	config #	-
Non-Sequential Object Position X	11	surface #	-	object #
Non-Sequential Object Position Y	12	surface #	-	object #

Non-Sequential Object Position Z	13	surface #	-	object #
Non-Sequential Object Tilt X	14	surface #	-	object #
Non-Sequential Object Tilt Y	15	surface #	-	object #
Non-Sequential Object Tilt Z	16	surface #	-	object #
Non-Sequential Object Parameters	17	surface #	parameter #	object #

GETZERNIKE

Calculates Zernike Fringe, Standard, or Annular coefficients for the currently loaded lens file, and places them in one of the vector arrays (either VEC1, VEC2, VEC3, or VEC4).

Syntax:

```
GETZERNIKE maxorder, wave, field, sampling, vector, zerntype, epsilon, reference
```

Discussion:

The maxorder argument is any number between 1 and 37 for Fringe or between 1 and 231 for Standard or Annular coefficients (see the discussion of zerntype below), and corresponds to the highest Zernike term desired. Wave and field are the integer values for the wavelength and field number respectively. The value for sampling determines the size of the grid used to fit the coefficients. Sampling may be 1 (32 x 32), 2 (64 x 64), etc.... up to 2048 x 2048. The vector argument must be an integer value between 1 and 4, and specifies which vector array the data should be placed in. The zerntype is 0 for "fringe" Zernike terms, 1 for "Standard" Zernike terms, and 2 for "Annular" Zernike terms. See "Zernike Fringe Coefficients", "Zernike Standard Coefficients", and "Zernike Annular Coefficients" for descriptions of these different types of Zernike terms. For Annular Zernike Coefficients epsilon is the annular ratio; this value is ignored for other Zernike types. To reference the OPD to the chief ray, the reference value should be zero or omitted; use 1 to reference to the surface vertex. If any of the arguments fall outside the valid ranges, then the nearest acceptable value is used instead.

The data is returned in one of the vector arrays with the following format: Vector position 1: Peak to valley in waves; Vector position 2: RMS to the zero OPD line in waves (this value is not physically meaningful but is provided for reference); Vector position 3: RMS to the chief ray in waves; Vector position 4: RMS to the image centroid in waves (this is the most physically meaningful number related to image quality); Vector position 5: Variance in waves; Vector position 6: Strehl ratio; Vector position 7: RMS fit error in waves; Vector position 8: Maximum fit error (at any one point) in waves. The remaining vector positions contain the actual Zernike coefficient data. For example, Zernike term number 1 is in vector position 9, Zernike term 2 is in position 10, and so on.

Example:

```
! This macro computes the first 37 Zernike Fringe coefficients
! for the currently loaded lens, at wave 1, field 1
! and a 32x32 grid density (sampling = 1). The coefficients
! will be placed in vector 1. First get the data:
GETZERNIKE 37,1,1,1,1,0
! Now print it out:
FORMAT 16.6
PRINT "Peak to Valley  : ", vec1(1)
PRINT "RMS to chief    : ", vec1(3)
PRINT "RMS to centroid : ", vec1(4)
PRINT "Variance        : ", vec1(5)
PRINT "Strehl ratio    : ", vec1(6)
PRINT "RMS Fit Error   : ", vec1(7)
PRINT "Maximum Fit Error : ", vec1(8)
i = 1
label 1
     FORMAT 2.0
     PRINT "Zernike #", i, " = ",
     FORMAT 16.6
     PRINT vec1(8+i)
```

```
      i = i + 1
 if (i < 38) THEN GOTO 1
 PRINT "All Done!"
```

GLAS
This command is obsolete. See "SETSURFACEPROPERTY, SURP".

GLASSTEMPLATE
GLASSTEMPLATE sets data on the Glass Substitution Template.
Syntax:
```
 GLASSTEMPLATE code, data
```

Discussion:
For checkbox values, code is 0 for Use Glass Substitution Template, 1 for Exclude Glasses With Incomplete Data, 11 for Standard, 12 for Preferred, 13 for Obsolete, 14 for Special, and 21-26 for Use Relative Cost, CR, FR, SR, AR, and PR, respectively. The checkbox will be unchecked if the value of data is zero. Any non-zero value will check the checkbox. For numerical values, code is 31-36 for Relative Cost, CR, FR, SR, AR, and PR, respectively.
Related Function:
GTEM

GLENSNAME
 GLENSNAME will place the current lens name at the top left corner of the text box on user defined graphics screens.
Syntax:
```
 GLENSNAME
```

Discussion:
GLENSNAME is primarily used for making your graphics look like other Zemax graphics.
Example:
See the section GRAPHICS.

GLOBALTOLOCAL
Converts coordinate breaks from global to local references.
Syntax:
```
 GLOBALTOLOCAL surf1, surf2, direction
```

Discussion:
GLOBALTOLOCAL concatenates all adjacent coordinate breaks into single coordinate breaks. Surf1 and surf2 define the surface range. Direction is 0 for forward, 1 for reverse coordinate breaks. See "Global To Local". See also "LOCALTOGLOBAL".

GOSUB, SUB, RETURN, and END
These four keywords are used together to define and call subprograms within the ZPL macro file. Each keyword has a special purpose. GOSUB is used to direct the program flow to a defined subroutine. SUB is used to define the subroutine name, as well as indicate the beginning of the subroutine body. RETURN indicates the macro execution should continue at the point where the most recent GOSUB call was placed. END indicates that the macro should terminate immediately.
Syntax:
See the Example section for sample syntax.
Discussion:
There can be no more than 100 defined subroutines per ZPL macro file. Each subroutine must be terminated by a RETURN command. More than one return command may be placed in the body of a subroutine. If subroutines are defined, at least one END command must be used to indicate the end of the main macro body. The main macro body must be at the top of the file.
There can be no more than 100 "nesting levels" used in the ZPL macro. For example, if subroutine ABC calls subroutine XYZ, then the nesting level is 2. If subroutine XYZ then calls subroutine DEF, the nesting level is 3.

All variables in ZPL are global. Any variables used or defined within a subroutine exist within the main macro as well.

Example:
```
x = 1 y = 2
GOSUB add
print "the sum of ", x, " and ", y, " is ", z
END

SUB add
z = x + y
RETURN
```

GO TO

Normally, each macro line is executed in turn. GOTO allows execution to resume at an arbitrary point in the macro. GOTO is always used in conjunction with the LABEL command.

Syntax:
```
GOTO label_number
GOTO text_label
```

Discussion:

There must be a LABEL command with the corresponding label_number or text_label somewhere in the macro, or an error will result.

Example:
```
LABEL 1
x = RAND(10)
if x <= 5 THEN GOTO 1
PRINT " X is greater than 5 "
```

GRAPHICS

Creates a standard Zemax graphics frame with ruling lines for the plot title.

For an easy way of creating plots, see "PLOT" and "PLOT2D".

Syntax:
```
GRAPHICS
GRAPHICS NOFRAME
GRAPHICS OFF
```

Discussion:

If GRAPHICS is specified alone, then a standard Zemax graphics window will be created. If the optional argument NOFRAME is supplied, then the standard frame for the graph title will be suppressed. All subsequent graphics commands will be sent to this newly created window. GRAPHICS OFF will close any existing open graphics windows, and then display the closed window.

Example:
```
graphics
xmx = xmax()
xmn = xmin()
ymx = ymax()
ymn = ymin()
xwidth = xmx-xmn
ywidth = ymx-ymn
xleft = xmn + ( .1 * xwidth )
xrigh = xmn + ( .9 * xwidth )
ytopp = ymn + ( .1 * ywidth )
ybott = ymn + ( .7 * ywidth )
line xleft,ytopp,xrigh,ytopp
line xrigh,ytopp,xrigh,ybott
line xrigh,ybott,xleft,ybott
line xleft,ybott,xleft,ytopp
gtitle " the rain in spain falls mainly on the plain"
glensname
gdate
```

```
gtext xmx/2,ymx/2,0, " start this text in the center."
gtextcent ymx*.05, " center this text near the top."
gtext xmx*.05,ymx*.75,90, " place me vertically near left edge."
gtext xmx*.15,ymx*.68,30, " orient me at 30 degrees."
graphics off
```

Related Keywords:
LINE, GITITLE, GLENSNAME, GDATE, GTEXTCENT, GTEXT, PIXEL, PLOT

GTEXT

GTEXT is used for labeling graphics plots with user defined text.
Syntax:
```
GTEXT x, y, angle, user_text
GTEXT x, y, angle, A$
```

Discussion:
The coordinates x and y refer to the left edge of where the text string user_text will appear. "user_text" may be either a constant string in quotes or a string variable name. Angle specifies how the text is rotated with respect to the graphics frame, and defaults to 0 degrees (horizontal). See also SETTEXTSIZE.
Example:
See "GRAPHICS".

GTEXTCENT

GTEXT is used for centering labels on graphics plots with user defined text.
Syntax:
```
GTEXTCENT y, user_text
```

Discussion:
The coordinate y refers to the vertical position of the text string user_text. See also SETTEXTSIZE.
Example:
See "GRAPHICS".

GTITLE

GTITLE is similar to GTEXT except only the text needs to be specified, and the text will appear centered in the title bar on the graphics display. GTITLE is useful for making ZPL graphics functions look like standard Zemax graphics.
Syntax:
```
GTITLE user_title
```

Example:
See "GRAPHICS".

HAMMER

Invokes the Hammer optimization algorithm to optimize the current lens with the current merit function.
Syntax:
```
HAMMER
HAMMER number_of_cycles
HAMMER number_of_cycles, algorithm
```

Discussion:
If no argument is provided, then the Hammer Optimization runs 1 cycle using Damped Least Squares. If an argument is provided, it must evaluate to an integer between 1 and 99, and the Hammer optimization algorithm will run the specified number of cycles. For the algorithm argument, use 0 for Damped Least Squares (the default) and 1 for Orthogonal Descent. For more information see "Performing an optimization".
Related Functions:
MFCN
Example:
```
PRINT "Starting merit function:", MFCN()
```

```
HAMMER 3
PRINT "Ending merit function :", MFCN()
```

IF-THEN-ELSE-ENDIF
IF provides conditional macro execution and branching.
Syntax:
```
IF (expression)
(commands)
ELSE (commands)
ENDIF
```

or
```
IF (expression) THEN (command)
```
Discussion:
The IF-ELSE-ENDIF construction is used for conditional execution of either the group of commands following the IF command or the commands following the ELSE command, but not both. The value of expression is considered false if it is zero, otherwise it is considered true. The expression can be any valid ZPL expression, composed of functions, variables, operands, and constants. The IF command must be paired with an ENDIF, although the ELSE is optional. IF-ENDIF pairs may be nested to any level.
The IF-THEN construction is handy for conditional execution of a single instruction. If the THEN keyword is specified, the IF command is terminated, and there is no need for an ENDIF. The ELSE keyword is not supported in the IF-THEN construction.
Example:
```
x = 1
y = 2
if (x < y)
     PRINT "x is less than y"
ELSE
     if (x == y) THEN PRINT "x equals y"
     if (x > y) THEN PRINT "x is greater than y"
ENDIF
```

IMA
Computes the Geometric Image Analysis feature and saves the result to a BIM file. For a description of the BIM (Binary Image) format see "The BIM format". For a description of the Geometric Image Analysis feature see "Geometric Image Analysis".
Syntax:
```
IMA outfilename, infilename
```

Discussion:
This keyword requires the name of the output BIM file, and optionally, the name of the input IMA or BIM file. If the extension to the outfilename is not provided, the extension BIM will be appended. The extension must be provided on the infilename. The filenames must be enclosed in quotes if any blank or other special characters are used. The outfilename will be placed in the <data>\<images> folder. The infilename must also be placed in the <data>\<images> folder. No paths should be provided with the file names.
The settings for the Geometric Image Analysis feature will be those settings previously saved for the current lens. To make adjustments to the settings, open a Geometric Image Analysis window, choose the appropriate settings, then press "Save". Be certain to choose "Show As" as anything other than "spot diagram". All subsequent calls to IMA will use the saved settings. The exceptions are the output file name, which is specified as the first argument after the IMA keyword, and the input source file, which is optionally specified as the second argument after the IMA keyword.
Example:
```
IMA "output.BIM", "LETTERF.IMA"
```

Related Keywords:
IMASHOW, IMASUM

IMAGECOMBINE

Combines two graphic files and writes the combined graphic to a new file.

Syntax:
```
IMAGECOMBINE source1$, source2$, destination$, mode$
```

Discussion:

This keyword opens the two source graphic files, combines them to a single image, and then writes the resulting image to the destination file. The files may be in BMP, JPG or PNG format. Conversion between the formats is automatic and is based upon the extension provided. If the source images have the same number of rows, they can be combined left-to-right. If the source images have the same number of columns, they can be combined top-to-bottom. The combination used depends upon the mode$ value. Use either "LEFTRIGHT" or "TOPBOTTOM" for the value of mode$ to define how the images should be combined.

Example:
```
S1$ = "C:\TEMP\SOURCE1.JPG" S2$ = "C:\TEMP\SOURCE2.JPG"
D$ = "C:\TEMP\DESTINATION.JPG" M$ = "LEFTRIGHT"
IMAGECOMBINE S1$, S2$, D$, M$
```

Related Functions:
PIXX, PIXY
Related Keyword:
IMAGEEXTRACT

IMAGEEXTRACT

Extracts a rectangular region of a graphic file and writes the portion of the graphic to a new file.

Syntax:
```
IMAGEEXTRACT source$, destination$, startx, starty, width, height
```

Discussion:

This keyword opens the source graphic, and extracts a rectangular portion of the file, then writes the extracted portion to the destination file. The files may be in BMP, JPG or PNG format. Conversion between the formats is automatic and is based upon the extension provided. The startx and starty values are the offsets from the left side and top side respectively, in pixels. The top row and left column start at number 1. The width and height values are the size in pixels of the rectangular region to extract.

Example:
```
S$ = "C:\TEMP\SOURCE.JPG"
D$ = "C:\TEMP\DESTINATION.JPG"
IMAGEEXTRACT S$, D$, 1, 1, 20, 20
```

Related Functions:
PIXX, PIXY
Related Keywords:
IMAGECOMBINE

IMASHOW

Displays an IMA or BIM file in a viewer window. For a description of the IMA and BIM file formats, see "The IMA format" and "The BIM format".

Syntax:
```
IMASHOW filename.ima
```

Discussion:

This keyword requires the name of the IMA or BIM file. The extension must be included. The filename may be enclosed in quotes if any blank or other special characters are used. The file must be located in the <data>\<im- ages> folder. This command will open a new window to display the file.

Example:
```
IMASHOW "LETTERF.IMA"
```

Related Keywords:

IMA, IMASUM

IMASUM

Sums the intensity in a BIM file. Despite the name of this function, currently only BIM files, and not IMA files, may be summed. This limitation is due to the low number of grey scales that are supported in the IMA file format. For a description of the IMA and BIM file formats, see "The IMA format" and "The BIM format".
Syntax:
```
IMASUM filename1, filename2, outfilename
```

Discussion:
This keyword requires the names of three BIM files (which need not be distinct). The BIM extension must be provided on all three file names. The filenames may be enclosed in quotes if any blank or other special characters are used. The files must be located in the <data>\<images> folder, which is where the output file will be. The two source files must have the same number of pixels and color channels, or an error message will result.
Example:
```
IMASUM "A.BIM", "B.BIM", "sum of A and B.BIM"
```

Related Keywords:
IMA, IMASHOW

IMPORTEXTRADATA

Imports data into the lens data editor from a file.
Syntax:
```
IMPORTEXTRADATA surface, filename
```

Discussion:
The surface may be any valid surface number that uses extra data values. The filename should include the full path. For the supported data formats, see "Importing grid data" under the sequential surface "Grid Sag".

INPUT

INPUT provides a means for prompting the user for numeric or text data when the macro is run.
Syntax:
```
INPUT "Prompt String" , variable
INPUT variable
INPUT "Prompt String" , string_variable$
INPUT string_variable$
```

Discussion:
The variable may be any valid variable name. If the variable name is a string variable, then the input will be interpreted as a literal string; otherwise, as a numeric. The INPUT command will use a "?" prompt if no prompt string is supplied. The prompt is always displayed on the screen, and the input always is accepted from the keyboard only. The prompt string cannot exceed 200 characters in length.
Example:
```
INPUT "Enter value for x:", x
PRINT "X = ", x
INPUT "Enter a value for A$:", A$
PRINT A$
```

INSERT

INSERT inserts a new surface in the lens data editor.
Syntax:
```
INSERT surf
```

Discussion:
The value surf must evaluate to an integer expression. See also DELETE and SURFTYPE.
Example:

```
INSERT 5
INSERT i+2*j
```

INSERTCONFIG

INSERTCONFIG inserts a new configuration in the multi-configuration editor.

Syntax:
```
INSERTCONFIG config
```

Discussion:
The value config must evaluate to an integer expression greater than 0 and less than or equal to the current number of configurations plus 1. See also DELETECONFIG and INSERTMCO.

Example:
```
INSERTCONFIG 4
```

INSERTMCO

INSERTMCO inserts a new multi-configuration operand in the multi-configuration editor.

Syntax:
```
INSE
RTMC
O
row
```

Discussion:
The value row must evaluate to an integer expression greater than 0 and less than or equal to the current number of operands plus 1. See also DELETEMCO and INSERTCONFIG.

Example:
```
INSERTMCO 7
```

INSERTMFO

INSERTMFO inserts a new merit function operand in the merit function editor.

Syntax:
```
INSE
RTMF
O
row
```

Discussion:
The value row must evaluate to an integer expression greater than 0 and less than or equal to the current number of operands plus 1. See also DELETEMFO and SETOPERAND.

Example:
```
INSE
RTMF
O 23
```

INSERTOBJECT

Inserts a new NSC object.

Syntax:
```
INSE
RTOB
JECT
surf
,
```

```
obje
ct
```

Discussion:
The value surf must evaluate to an integer expression, and the specified surface must be a non-sequential component surface. Use a surf value of 1 if the program mode is non-sequential. The value object must evaluate to an integer that is between 1 and the current number of objects plus 1, inclusive. The new object will be inserted at the specified object number, renumbering other objects as required. See also DELETEOBJECT and SETN- SCPROPERTY.
Example:

```
INSE
RTOB
JECT
1, 1
```

INSERTTOL
INSERTTOL inserts a new operand in the tolerance data editor.
Syntax:

```
INSE
RTTO
L
row
```

Discussion:
The value row must evaluate to an integer expression greater than 0 and less than or equal to the current number of operands plus 1. See also DELETETOL and SETTOL.
Example:

```
INSE
RTTO
L 3
```

LABEL
LABEL provides a destination for the GOTO command, see "GOTO" for details.
Syntax:
```
LABEL label_number
LABEL text_label
```

Discussion:
The label_number must be an integer value greater than zero, such as 1 or 7. If a text label is used, the text label must not contain spaces or other special characters that are used as delimiters. LABEL has no effect on program flow by itself. There is a limit of 300 label commands in a macro.
Example:
```
LABEL 7
LABEL startover
```

LINE
LINE is the primitive line drawing function for graphical displays.
Syntax:
```
LINE oldx, oldy, newx, newy
```

Discussion:
LINE will evaluate the four expressions and draw a line connecting the points defined. The coordinates refer to the current graphics frame, and must be contained within the bounds defined by XMIN, YMIN, XMAX, and YMAX. Although only integer pixel values can actually be plotted, LINE will accept real values as arguments and round the coordinates to the nearest integer equivalents. LINE is only valid in graphics mode. The color of the line drawn is controlled by the current pen color, which is specified using the COLOR keyword.
Example:

See "GRAPHICS".

LOADARCHIVE

Opens an existing Zemax archive (*.ZAR) file.

Syntax:

```
LOADARCHIVE filename, extractpath
```

Discussion:

The filename must include the name of the archive file including the file extension. If the filename does not include a complete path to the archive file, the default folder for lenses is assumed. If no extractpath is defined, the path is assumed to be the same as the filename. The lens and session files defined in the archive file are then opened. See "Restore From Archive File".

LOADCATALOG

Reloads glass and coating catalogs for the currently loaded lens.

Syntax:

```
LOADCATALOG
```

Discussion:

When lenses are loaded, any associated glass catalogs and data files, including the COATING.DAT file, are automatically loaded if they are not already loaded. However, if these catalogs have been modified, perhaps by the ZPL macro itself; then the LOADCATALOG keyword may be used to force a reload of the catalogs. Use of this keyword is not required unless the COATING.DAT or glass AGF catalog files have been modified since the start of the current Zemax session.

LOADDETECTOR

Loads the data saved in a file to an NSC Detector Rectangle, Detector Color, Detector Polar, or Detector Volume object.

Syntax:

```
LOADDETECTOR surf, object, filename
```

Discussion:

This keyword requires numeric expressions that specify the surface and object, and a file name with or without a full path. Surf is the surface number of the non-sequential group; use 1 when using non-sequential mode. Object is the object number of the detector object. The filename may include the full path, if no path is provided the path of the current lens file is used. The extension should be DDR, DDC, DDP, or DDV for Detector Rectangle, Color, Polar, and Volume objects, respectively.

Related Keywords:

SAVEDETECTOR

LOADLENS

Loads a new lens file, replacing any lens file currently in memory.

Syntax:

```
LOADLENS filename, appendflag, session
```

Discussion:

LOADLENS will load a lens file. If the filename contains the complete path, such as C:\MYDIR\MYLENS.ZMX, then the specified file will be loaded. If the path is left off, then the default folder for lenses will be used (see "Folders").

If the appendflag is zero or absent, then LOADLENS loads the file. If the appendflag is greater than zero, then the file is appended to the current lens starting at the surface specified by the value of the appendflag. The appendflag should only be used when appending one sequential system to another. Appending non-sequential systems isn't currently supported.

If the session flag is non-zero, any associated session file will be loaded with the lens and all windows will be updated otherwise the lens session file is ignored.

Example:

```
LOADLENS "COOKE.ZMX"
```

Related Keywords:

SAVELENS

LOADMERIT
Loads a merit function file, replacing the merit function in the current lens.
Syntax:
```
LOADMERIT "filename"
LOADMERIT file$
```

Discussion:
LOADMERIT will load a new merit function from a .MF file. If the filename contains the complete path, such as C:\MYDIR\MYLENS.MF, then the specified file will be loaded. If the path is left off, then the <data>\MeritFunc- tion folder will be used (see "Folders"). This keyword can also load just the merit function from a ZMX file using the same syntax. See also SAVEMERIT.

LOADTOLERANCE
Loads a tolerance file, replacing the tolerances in the current lens.
Syntax:
```
LOADTOLERANCE "filename"
LOADTOLERANCE file$
```

Discussion:
LOADTOLERANCE will load a new tolerance file. If the filename contains the complete path, such as C:\MYDIR\MYLENS.TOL, then the specified file will be loaded. If the path is left off, then the <data>\Tolerance folder will be used (see "Folders"). See also SAVETOLERANCE.

LOCALTOGLOBAL
Converts coordinate breaks from local to global references.
Syntax:
```
LOCALTOGLOBAL surf1, surf2, reference
```

Discussion:
LOCALTOGLOBAL modifies all coordinate breaks into groups of 3 coordinate breaks, providing a means to locate surfaces in global coordinates. Surf1 and surf2 define the surface range. Reference is the reference surface number, which must precede s1. See "Local To Global". See also "GLOBALTOLOCAL".

LOCKWINDOW
Locks any one or all open windows.
Syntax:
```
LOCKWINDOW winnum
```
Discussion:
See "Graphic windows operations". If the winnum is zero, then all open windows are locked. If the winnum argument is -1, then the currently executing window will lock at the end of the macro execution.
Example:
```
LOCKWINDOW 7
```

Related Keywords:
UNLOCKWINDOW

MAKEFACETLIST
Makes an ASCI listing of the vertices of a faceted description of any importable CAD file.
Syntax:
```
MAKEFACETLIST infile, outfile
```

Discussion:

The infile is a string variable or name with the full path to a CAD format file. The outfile is a string variable or name to the full path for the output file. The output file will be a text listing. There are 18 lines for each triangle in the faceted description of the object. The first 6 lines are data for vertex 1 of the first triangle in this format:

```
x coordinate
y coordinate
z coordinate
x normal vector
y normal vector
z normal vector
```

The next 6 values are for vertex 2, and then 6 more values for vertex 3. The 18 values together define the first triangle. The pattern repeats for the next triangle; each represented by 18 values, until all the triangles are written.

MAKEFOLDER

Makes a folder for files.

Syntax:

```
MAKEFOLDER "C:\TEMP\TEST_FOLDER"
MAKEFOLDER path$
```

Discussion:

The new folder will be created. If the folder already exists the keyword is ignored. If the path is invalid, an error message is issued and the macro execution will be terminated.

MODIFYSETTINGS

Modifies data within the settings files used by GETTEXTFILE.

Syntax:

```
MODIFYSETTINGS settingsfilename, type, value
```

Discussion:

The settingsfilename must be in quotes, or be a string variable name, and include the full path, name, and extension for the file to be modified.

The type argument is a text mnemonic that indicates which setting within the file is to be modified. The supported values for the type argument are listed in the table below. Because there are many different types of analysis windows, and each has many different settings available, the list of types does not include all possible settings. More type values may be added upon request; please contact technical support for assistance.

Type codes supported by MODIFYSETTINGS

Feature	Available type codes
2D Layout	LAY_RAYS: The number of rays.
Detector Viewer	DVW_SURFACE: The surface number. Use 1 for Non-Sequential mode. DVW_DETECTOR: The detector number.
	DVW_SHOW: The "show as" setting. The meaning depends upon the type of window displayed:
	For Graphics Windows: Use 0 for grey scale, 1 for inverted grey scale, 2 for false color, 3 for inverted false color, 4 for cross section row, and 5 for cross section column.
	For Text Windows: Use 0 for full pixel data, 1 for cross section row, and 2 for cross section column.
	DVW_ROWCOL: The row or column number for cross section plots. DVW_ZPLANE: The Z-Plane number for detector volumes.

	DVW_SCALE: The scale mode. Use 0 for linear, 1 for Log -5, 2 for Log -10, and
	3 for Log - 15.
	DVW_SMOOTHING: The integer smoothing value.
	DVW_DATA: Use 0 for incoherent irradiance, 1 for coherent irradiance, 2 for coherent phase, 3 for radiant intensity, 4 for radiance (position space), and 5 for radiance (angle space).
	DVW_ZRD: The ray data base name, or null for none. DVW_FILTER: The filter string.
	DVW_MAXPLOT: The maximum plot scale. DVW_MINPLOT: The minimum plot scale. DVW_OUTPUTFILE: The output file name.
Extended Diffraction Image Analysis	EXD_DISPLAYSIZE: The display size. EXD_FIELD: The field number. EXD_FILESIZE: The file size. EXD_WAVE: The wavelength number.
FFT Line/Edge Spread	LSF_COHERENT: Use 0 for incoherent, 1 for coherent
	LSF_TYPE: Use 0-9 for X-Linear, Y-Linear, X-Log, Y-Log, X-Phase, Y-Phase, X- Real, Y-Real, X-Imaginary, or Y-Imaginary, respectively.
	LSF_SAMP: The sampling, use 1 for 32 x 32, 2 for 64 x 32, etc. LSF_SPREAD: Use 0 for line, 1 for edge.
	LSF_WAVE: The wavelength number, use 0 for polychromatic (incoherent only) LSF_FIELD: The field number.
	LSF_POLARIZATION: Use 0 for unpolarized, 1 for polarized.
	LSF_PLOTSCALE: The plot scale.
FFT PSF	PSF_TYPE: Use 0-4 for Linear, Log, Phase, Real, or Imaginary, respectively. PSF_SAMP: The sampling, use 1 for 32 x 32, 2 for 64 x 32, etc.
	PSF_WAVE: The wavelength number, use 0 for polychromatic.
	PSF_FIELD: The field number.
	PSF_SURFACE: The surface number, use 0 for image. PSF_POLARIZATION: Use 0 for unpolarized, 1 for polarized. PSF_NORMALIZE: Use 0 for unnormalized, 1 for unity normalization. PSF_IMAGEDELTA: The image point spacing in micrometers.
FFT PSF Cross Section	PSF_TYPE: Use 0-9 for X-Linear, Y-Linear, X-Log, Y-Log, X-Phase, Y-Phase, X- Real, Y-Real, X-Imaginary, or Y-Imaginary, respectively.
	PSF_ROW: The row number (if doing an X scan) or column number (if doing a y scan). Use 0 for center.
	PSF_SAMP: The sampling, use 1 for 32 x 32, 2 for 64 x 32, etc. PSF_WAVE: The wavelength number, use 0 for polychromatic. PSF_FIELD: The field number.

	PSF_POLARIZATION: Use 0 for unpolarized, 1 for polarized.
	PSF_NORMALIZE: Use 0 for unnormalized, 1 for unity normalization. PSF_PLOTSCALE: The plot scale.
Footprint Diagram	FOO_RAYDENSITY: The ray density. Use 0 for ring, 1 for 10, 2 for 15, 3 for 20 etc.
	FOO_SURFACE: The surface number. FOO_FIELD: The field number. FOO_WAVELENGTH: The wavelength number.
	FOO_DELETEVIGNETTED: Delete vignetted, use 0 for no, 1 for yes.
Geometric Bitmap Image Analysis	GBM_FIELDSIZE: The field Y size.
	GBM_RAYS: The number of rays per source pixel. GBM_XPIX: The number of X pixels.
	GBM_YPIX: The number of Y pixels. GBM_XSIZ: The X pixel size. GBM_YSIZ: The Y pixel size. GBM_INPUT: The input file name GBM_OUTPUT: The output file name GBM_SURFACE: The surface number GBM_ROTATION: The rotation setting
Geometric Image Analysis	IMA_FIELD: The field size. IMA_IMAGESIZE: The image size. IMA_IMANAME: The image file name. IMA_KRAYS: The number of rays x 1000. IMA_NA: The numerical aperture. IMA_OUTNAME: The output file name. IMA_SURFACE: The surface number. IMA_PIXELS: The number of pixels.
FFT Through Focus MTF	TFM_SAMP: The sampling. Use 1 for 32x32, 2 for 64x64, etc. TFM_DELTAFOC: The delta focus.
	TFM_FREQ: The spatial frequency for which the data is plotted. TFM_STEPS: The number of focal plane steps.
	TFM_WAVE: The wavelength number. Use 0 for all. TFM_FIELD: The field number. Use 0 for all.
	TFM_TYPE: The data type. Use 0 for modulation, 1 for real, 2 for imaginary, 3 for phase, or 4 for square wave.
	TFM_POLAR: Use polarization. Use 0 for no, 1 for yes. TFM_DASH: Use dashes. Use 0 for no, 1 for yes.
Huygens MTF	HMF_PUPILSAMP: The pupil sampling. Use 1 for 32x32, 2 for 64x64, etc. HMF_IMAGESAMP: The image sampling. Use 1 for 32x32, 2 for 64x64, etc. HMF_IMAGEDELTA: The image point spacing in micrometers. HMF_CONFIG: The configuration number. Use 0 for all, 1 for current, etc. HMF_WAVE: The wavelength number. Use 0 for polychromatic. HMF_FIELD: The field number. Use 0 for all.
	HMF_TYPE: The data type. Currently only modulation (0) is supported.
	HMF_MAXF: The maximum spatial frequency. HMF_POLAR: Use polarization. Use 0 for no, 1 for yes. HMF_DASH: Use dashes. Use 0 for no, 1 for yes.

Huygens Through Focus MTF	HTF_PUPILSAMP: The pupil sampling. Use 1 for 32x32, 2 for 64x64, etc. HTF_IMAGESAMP: The image sampling. Use 1 for 32x32, 2 for 64x64, etc. HTF_IMAGEDELTA: The image point spacing in micrometers. HTF_CONFIG: The configuration number. Use 0 for all, 1 for current, etc. HTF_FREQ: The spatial frequency for which data is plotted. HTF_WAVE: The wavelength number. Use 0 for all. HTF_FIELD: The field number. Use 0 for all. HTF_TYPE: The data type. Currently only modulation (0) is supported. HTF_DELTAFOC: The delta focus. HTF_STEPS: The number of focal plane steps. HTF_POLAR: Use polarization. Use 0 for no, 1 for yes. HTF_DASH: Use dashes. Use 0 for no, 1 for yes.
Huygens MTF vs. Field	HMH_SAMP: The sampling. Use 1 for 32x32, 2 for 64x64, etc. HMH_SCANTYPE: The field scan type. Use 0 for +Y, 1 for +X, etc. HMH_WAVE: The wavelength. Use 0 for all. HMH_FIELDDENSITY: The field density. HMH_FREQ1: Spatial frequency 1. HMH_FREQ2: Spatial frequency 2. HMH_FREQ3: Spatial frequency 3. HMH_FREQ4: Spatial frequency 4. HMH_FREQ5: Spatial frequency 5. HMH_FREQ6: Spatial frequency 6. HMH_POLAR: Use polarization. Use 0 for no, 1 for yes. HMH_DASH: Use dashes. Use 0 for no, 1 for yes. HMH_REMOVEVIGNETTING: Remove vignetting factors. Use 0 for no, 1 for yes.
Huygens PSF	HPS_PUPILSAMP: The pupil sampling, use 1 for 32 x 32, 2 for 64 x 64, etc. HPS_IMAGESAMP: The image sampling, use 1 for 32 x 32, 2 for 64 x 64, etc. HPS_WAVE: The wavelength number, use 0 for polychromatic. HPS_FIELD: The field number. HPS_IMAGEDELTA: The image point spacing in micrometers. HPS_TYPE: The data type. Use 0-8 for Linear, Log -1, Log -2, Log -3, Log -4, Log -5, Real, Imaginary, or Phase, respectively.
Huygens PSF Cross Section	HPC_PUPILSAMP: The pupil sampling, use 1 for 32 x 32, 2 for 64 x 64, etc. HPC_IMAGESAMP: The image sampling, use 1 for 32 x 32, 2 for 64 x 64, etc. HPC_WAVE: The wavelength number, use 0 for polychromatic. HPC_FIELD: The field number. HPC_IMAGEDELTA: The image point spacing in micrometers. HPC_TYPE: The data type. Use 0-9 for X-Linear, X-Log, Y-Linear, Y-Log, X- Real, Y-Real, X-Imaginary, Y-Imaginary, X-Phase, or Y-Phase, respectively.
Illumination XY	ILL_SOURCE: The source size. ILL_SMOOTH: The smoothing value to use.

Scan	ILL_DETSIZE: The detector size. ILL_SURFACE: The surface number.
Image Simulation	ISM_INPUTFILE: The input file name. This should be specified without a path. ISM_FIELDHEIGHT: The Y field height.
	ISM_OVERSAMPLING: Oversample value. Use 0 for none, 1 for 2X, 2 for 4x, etc.
	ISM_GUARDBAND: Guard band value. Use 0 for none, 1 for 2X, 2 for 4x, etc. ISM_FLIP: Flip Source. Use 0 for none, 1 for TB, 2 for LR, 3 for TB&LR. ISM_ROTATE: Rotate Source: Use 0 for none, 1 for 90, 2 for 180, 3 for 270. ISM_WAVE: Wavelength. Use 0 for RGB, 1 for 1+2+3, 2 for wave #1, 3 for wave
	#2, etc.
	ISM_FIELD: Field number.
	ISM_PSAMP: Pupil Sampling. Use 1 for 32x32, 2 for 64x64, etc. ISM_ISAMP: Image Sampling. Use 1 for 32x32, 2 for 64x64, etc. ISM_PSFX, ISM_PSFY: The number of PSF grid points. ISM_ABERRATIONS: Use 0 for none, 1 for geometric, 2 for diffraction. ISM_POLARIZATION: Use 0 for no, 1 for yes. ISM_FIXEDAPERTURES: Use 0 for no, 1 for yes.
	ISM_USERI: Use 0 for no, 1 for yes.
	ISM_SHOWAS: Use 0 for Simulated Image, 1 for Source Bitmap, and 2 for PSF Grid.
	ISM_REFERENCE: Use 0 for chief ray, 1 for vertex, 2 for primary chief ray. ISM_SUPPRESS: Use 0 for no, 1 for yes.
	ISM_PIXELSIZE: Use 0 for default or the size in lens units.
	ISM_XSIZE, ISM_YSIZE: Use 0 for default or the number of pixels. ISM_FLIPIMAGE: Use 0 for none, 1 for top-bottom, etc. ISM_OUTPUTFILE: The output file name or empty string for no output file.
MTF - FFT	MTF_SAMP: The pupil sampling, use 1 for 32, 2 for 64, etc. MTF_WAVE: The wavelength number, use 0 for all. MTF_FIELD: The field number, use 0 for all.
	MTF_TYPE: Use 0 for modulation, 1 for real, 2 for imaginary, 3 for phase, 4 for square wave.
	MTF_SURF: The surface number, use 0 for image. MTF_MAXF: The maximum frequency, use 0 for default. MTF_SDLI: Show diffraction limit, 0 for no, 1 for yes. MTF_POLAR: Polarization, 0 for no, 1 for yes. MTF_DASH: Use dashes, 0 for no, 1 for yes.
NSC Object Viewer	SHA_ROTX: The x rotation in degrees. SHA_ROTY: The y rotation in degrees. SHA_ROTZ: The z rotation in degrees.
NSC Shaded Model	SHA_ROTX: The x rotation in degrees. SHA_ROTY: The y rotation in degrees. SHA_ROTZ: The z rotation in degrees.
Partially	PCI_FIELD: The field number. PCI_FILESIZE: The file size. PCI_WAVE: The

Coherent Image Analysis	wavelength number.
	PCI_RESAMPLE: The resample image setting, 0 for no 1 for yes.
	PCI_RSNX: The resample number x PCI_RSNY: The resample number y PCI_RSDCX: The resample decenter x PCI_RSDCY: The resample decenter y PCI_RSDLX: The resample delta x PCI_RSDLY: The resample delta y
Polarization Pupil Map	PPM_SAMP: The sampling, use 0 for 3x3, 1 for 5x5, 2 for 7x7, etc. PPM_FIELD: The field number.
	PPM_WAVE: The wavelength number. PPM_SURFACE: The surface number. PPM_JX: The Jx amplitude.
	PPM_JY: The Jy amplitude. PPM_PX: The Px phase. PPM_PY: The Py phase.
	PPM_ADDCONFIG: The add configs string. PPM_SUBCONFIGS: The subtract configs string.
Physical Optics Propagation - General Tab	POP_END: The end surface. POP_FIELD: The field number. POP_START: The starting surface. POP_WAVE: The wavelength number.
Physical Optics Propagation - Beam Definition Tab	POP_AUTO: Simulates the pressing of the "auto" button which chooses appropriate X and Y beam widths based upon the sampling and other settings. POP_BEAMTYPE: Selects the beam type. Use 0 for Gaussian Waist, 1 for
	Gaussian Angle, 2 for Gaussian Size + Angle, 3 for Top Hat, 4 for File, 5 for DLL
	and 6 for Multimode.
	POP_PARAMn: Sets beam parameter n, for example, use POP_PARAM3 to set parameter3.
	POP_PEAKIRRAD: Sets the normalization by peak irradiance. POP_POWER: Sets the normalization by total beam power. POP_SAMPX: The X direction sampling, use 1 for 32, 2 for 64, etc. POP_SAMPY: The Y direction sampling, use 1 for 32, 2 for 64, etc.
	POP_SOURCEFILE: The file name if the starting beam is defined by a ZBF file, DLL, or multimode file.
	POP_WIDEX: The X direction width. POP_WIDEY: The Y direction width.
Physical Optics Propagation - Fiber Data Tab	POP_COMPUTE: Use 1 to check the fiber coupling integral on, 0 to check it off. POP_FIBERFILE: The file name if the fiber mode is defined by a ZBF or DLL. POP_FIBERTYPE: Use the same values as POP_BEAMTYPE above, except
	for multimode which is not yet supported.
	POP_FPARAMn: Sets fiber parameter n, for example, use POP_FPARAM3 to set fiber parameter3.

	POP_IGNOREPOL: Use 1 to ignore polarization, 0 to consider polarization. POP_POSITION: Fiber position setting. Use 0 for chief ray, 1 for surface vertex. POP_TILTX: The X-Tilt.
	POP_TILTY: The Y-Tilt.
Relative Illumination	REL_RAYDENSITY: The number of rays. REL_FIELDDENSITY: The number of field points. REL_WAVE: The wavelength number, use 0 for all.
	REL_POLAR: Use 1 to use polarization, 0 to ignore polarization
	REL_LOG: Use 1 for a log scale, 0 for linear.
	REL_REMOVEVIGNETTING: Use 1 to remove vignetting factors, otherwise 0. REL_SCANTYPE: Use 0 for +y, 1 for +x, 2 for -y, or 3 for -x scan direction.
Shaded Model	SHA_ROTX: The x rotation in degrees. SHA_ROTY: The y rotation in degrees. SHA_ROTZ: The z rotation in degrees.
Spot Diagram	SPT_RAYS: The ray density.
Surface Sag	SRS_SAMP: The sampling. Use 1 for 33x33, 2 for 65x65, etc. SRS_SURF: The surface number.
Universal Plot 1D	UN1_CATEGORY: Use 0 for surface, 1 for system, 2 for config. UN1_PARAMETER: Use 0 for first option, 1 for second option, etc. UN1_SURFACE: The surface or configuration number. UN1_STARTVAL: The start value for the independent variable. UN1_STOPVAL: The stop value for the independent variable. UN1_STEPS: The number of steps between start and stop. UN1_OPERAND: The optimization operand name.
	UN1_MFLINE: The optimization operand line number. Use 0 for MF value.
	UN1_PAR1: Operand parameter 1. UN1_PAR2: Operand parameter 2. UN1_PAR3: Operand parameter 3. UN1_PAR4: Operand parameter 4. UN1_PAR5: Operand parameter 5. UN1_PAR6: Operand parameter 6. UN1_PAR7: Operand parameter 7. UN1_PAR8: Operand parameter 8.
	UN1_PLOTMIN: The minimum plot value for the dependent variable.
	UN1_PLOTMAX: The maximum plot value for the dependent variable. UN1_TITLE: The plot title.
Universal Plot 2D	UN2_CATEGORYX: Use 0 for surface, 1 for system, 2 for config. UN2_PARAMETERX: Use 0 for first option, 1 for second option, etc. UN2_SURFACEX: The surface or configuration number. UN2_STARTVALX: The start value for the independent variable. UN2_STOPVALX: The stop value for the independent variable. UN2_STEPSX: The number of steps between start and stop. UN2_CATEGORYY: Use 0 for surface, 1 for system, 2 for config. UN2_PARAMETERY: Use 0 for first option, 1 for second option, etc. UN2_SURFACEY: The surface or configuration number. UN2_STARTVALY: The start value for the independent variable. UN2_STOPVALY: The stop value for the independent variable. UN2_STEPSY: The number of steps between start and stop.

	UN2_OPERAND: The optimization operand name.
	UN2_MFLINE: The optimization operand line number. Use 0 for MF value.
	UN2_PAR1: Operand parameter 1. UN2_PAR2: Operand parameter 2. UN2_PAR3: Operand parameter 3. UN2_PAR4: Operand parameter 4. UN2_PAR5: Operand parameter 5. UN2_PAR6: Operand parameter 6. UN2_PAR7: Operand parameter 7. UN2_PAR8: Operand parameter 8.
	UN2_SHOWAS: Data display. Use 0 for surface, 1 for contour, etc.
	UN2_CONTOURFORMAT: Contour format string.
	UN2_PLOTMIN: The minimum plot value for the dependent variable. UN2_PLOTMAX: The maximum plot value for the dependent variable. UN2_TITLE: The plot title.
Wavefront Map	WFM_SAMP: The sampling, use 1 for 32, 2 for 64, etc. WFM_FIELD: The field number.
	WFM_WAVE: The wavelength number. WFM_SUBSR: The sub aperture radius. WFM_SUBSX: The sub aperture X decenter. WFM_SUBSY: The sub aperture Y decenter.

The value parameter is the new data for the specified setting. The modified settings file is written back to the original settings fie name.
See also GETTEXTFILE.
Example:
```
MODIFYSETTINGS "C:\MySPT.CFG", SPT_RAYS, 24
MODIFYSETTINGS "C:\MyPOP.CFG", POP_SOURCEFILE, "MyStartBeam.ZBF"
```

NEXT
See "FOR, NEXT".

NSLT
Initiates a Non-Sequential LightningTrace.
Syntax:
```
NSLT surf, source, ray_sampling, edge_sampling
```

Discussion:
Surf is an integer value that indicates the number of the Non-Sequential surface. If the program mode is set to Non-Sequential, use 1. Source refers to the object number of the desired source. If source is zero, all sources will be traced. Ray_sampling refers to the resolution of the LightningTrace mesh, with valid values being between 0 (corresponding to a mesh resolution of "Low (1X)") and 5 (corresponding to a mesh resolution of "1024X").
Edge_sampling refers to the resolution used in refining the LightningTrace mesh near the edges of objects, with valid values being between 0 (corresponding to a mesh resolution of "Low (1X)") and 4 (corresponding to a mesh resolution of "256X"). For more details on LightningTrace, see "The LightningTrace Control".
NSLT always calls UPDATE before executing a LightningTrace to make certain all objects are correctly loaded and updated.
Example:
```
NSLT 1, 0, 3, 2
```

NSTR
Initiates a Non-Sequential trace.
Syntax:

```
NSTR surf, source, split, scatter, usepolar, ignore_errors, random_seed, save,
savefilename, filter, zrd_format
```

Discussion:
Surf is an integer value that indicates the number of the Non-Sequential surface. If the program mode is set to Non-Sequential, use 1. Source refers to the object number of the desired source. If source is zero, all sources will be traced. If Split is non-zero, then splitting is on, otherwise, ray splitting is off. If Scatter is non-zero, then scattering is on, otherwise scattering is off. If Usepolar is non-zero then polarization will be used, otherwise polarization is off. If splitting is on polarization is automatically selected. If ignore_errors is non-zero, then errors will be ignored, otherwise ray errors will terminate the non-sequential trace and macro execution and an error will be reported.

If random_seed is zero, then the random number generator will be seeded with a random value, and every call to NSTR will produce different random rays. If random_seed is any integer other than zero, then the random number generator will be seeded with the specified value, and every call to NSTR using the same seed will produce identical rays. When using NSTR for optimization, it is recommended that a non-zero value be used for random_seed.

If save is omitted or is zero, the arguments savefilename, filter, and zrd_format need not be supplied. If save is not zero, the rays will be saved to a file. The saved data file will have the name specified by the savefilename. The file naming convention and destination folder are the same as described in "Saving ray data to a file". The extension of savefilename should be provided, but no path should be specified. If save is not zero, then the optional filter name is either a string variable with the filter, or the literal filter in double quotes. If no filter is being used, enter an empty pair of double quotes like this: "". For information on filter strings see "The filter string". For ZRD files, the zrd_format can be 0, 1, or 2 for uncompressed full data, compressed basic data, or compressed full data, respectively. For more information on ZRD formats see "Ray database (ZRD) files".

NSTR always calls UPDATE before tracing rays to make certain all objects are correctly loaded and updated.

Related Functions:
NSDD
Example:
```
NSTR 1, 0, 0, 0, 0, 1, 0, 1, "saverays.ZRD", "h2"
```

NUMFIELD

This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

NUMWAVE

This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

OPEN

Opens an existing text file for reading by the READ command.
Syntax:
```
OPEN "filename"
OPEN A$
```

Discussion:
The filename provided must be a valid file within quotes, or a string variable name containing the file name. See the keywords READ and CLOSE. Always CLOSE a file after all the data has been read.
Related Functions:
EOFF
Related Keywords:
READ, READNEXT, CLOSE
Example:
```
PRINT "Reading the double-column file TEST.DAT!" OPEN "TEST.DAT"
READ x1, y1
READ x2, y2
READ x3, y3
CLOSE
```

OPENANALYSISWINDOW

Opens a new analysis window.

Syntax:
```
OPENANALYSISWINDOW type, settingsfilename
```

Discussion:

The type argument is a 3 character string code that indicates the type of analysis to be performed. For the full list of string codes, see the "String Codes" section of the The Programming Tab. If no string code is provided or recognized, no window is opened. The settingsfilename contains the name of the CFG settings file to use. If this argument is missing, the default settings will be used.

Related Functions:

WINL, WINN

Related Keywords:

CLOSEWINDOW, MODIFYSETTINGS

Example:
```
FORMAT 0.0
PRINT "There are ", WINN(), " Windows currently open."
OPENANALYSISWINDOW "mtf"
PRINT "Just opened window ", WINL(), "."
PAUSE TIME, 3000
PRINT "There are ", WINN(), " Windows currently open."
CLOSEWINDOW WINL()
PAUSE TIME, 1000
PRINT "There are ", WINN(), " Windows currently open."
```

OPTIMIZE

Invokes the optimization algorithm to optimize the current lens with the current merit function.

Syntax:
```
OPTIMIZE
OPTIMIZE number_of_cycles
OPTIMIZE number_of_cycles, algorithm
```

Discussion:

The expression for number_of_cyles must evaluate to an integer value between 1 and 99, and the optimization algorithm will run the specified number of cycles. If number_of_cycles evaluates to zero, then the optimization runs in "Automatic" mode, stopping when the algorithm detects the process has converged. For the algorithm argument, use 0 for Damped Least Squares (the default) and 1 for Orthogonal Descent. For more information see "Performing an optimization".

To update the merit function without optimizing, use the MFCN function.

Related Functions:

MFCN

Example:
```
PRINT "Starting merit function:", MFCN()
OPTIMIZE
PRINT "Ending merit function :", MFCN()
```

OPTRETURN

Used to return numerical values back to the optimization algorithm through the use of the ZPLM optimization operand.

Syntax:
```
OPTRETURN datafield, result
```

Discussion:

The datafield expression must evaluate to an integer between 0 and 50. The datafield refers to a position in an array where the value of the expression result may be stored. The sole purpose of OPTRETURN is to be able to optimize values computed within a ZPL macro.

The optimization operand ZPLM must be used in the merit function to call the ZPL macro and retrieve the value returned by OPTRETURN. See "Optimizing with ZPL macros" for details.

Example:
```
x = sqrt(thic(3) + radi(5))
OPTRETURN j, x+5
```

OUTPUT

Specifies destination for text output. Output is either to the screen, or to a file.

Syntax:
```
OUTPUT SCREEN
OUTPUT filename
OUTPUT filename, APPEND
```

Discussion:

If OUTPUT SCREEN is specified alone, then all subsequently executed PRINT commands will be directed to the screen. If a valid filename is provided, then subsequent PRINT commands will output to the filename specified. If a filename with no path is provided, output will be directed to the <data>\Macros folder. To close the file created earlier, use OUTPUT SCREEN which will direct subsequent PRINT outputs to the screen. SHOWFILE will close the file and send it to the text viewer program for screen display. PRINTFILE will close the file and print it on the currently defined printer. Note that when sending PRINT commands to an output file, the file will be in Unicode format. To convert the file to ANSI, first close the file (for example using OUTPUT SCREEN) and then use the CONVERTFILEFORMAT keyword.

If the keyword APPEND follows the file name, then subsequent output will be appended to the file. Otherwise, the contents of the file will be overwritten.

Example:
```
OUTPUT "x.txt"
PRINT "This will not appear on the screen, but in the file x.txt."
OUTPUT SCREEN
PRINT "This will appear on the screen."
OUTPUT "x.txt", APPEND
PRINT "This will appear after the first line in the file x.txt."
```

Related Keywords:
CLOSE, OPEN, SHOWFILE, PRINTFILE

PARM

This command is obsolete. See "SETSURFACEPROPERTY, SURP".

PARAXIAL

Used to control whether ray tracing is done with paraxial or real rays.

Syntax:
```
PARAXIAL ON
PARAXIAL OFF
```

Discussion:

The current paraxial mode can be established by a call to the function PMOD, which returns 0 if paraxial mode is off, 1 if it is on. This feature is used for switching between real and paraxial ray traces. Certain calculations, such as measuring distortion and computing first-order properties such as effective focal length, require the tracing of paraxial rays.

Example:
```
mode = PMOD()
IF mode THEN PRINT "Paraxial mode is on!"
IF !mode THEN PRINT "Paraxial mode is not on!"
PARAXIAL ON
PRINT "Now paraxial mode is on!"
PRINT "Restoring original mode..."
if !mode THEN PARAXIAL OFF
```

PAUSE

Pauses macro execution, optionally while displaying a status message. The status message can be a string or numerical value. The macro continues once the user presses the "OK" button on the status dialog.

Syntax:
```
PAUSE
PAUSE "Ready to continue..."
PAUSE TIME, time
PAUSE THREADS
```

Discussion:
This feature may be used for debugging, presenting results, or pausing the execution of the macro. If the keyword after the PAUSE command is TIME, then the macro execution will "sleep" for *at least* the number of milliseconds given by the value provided. For example, to pause the macro for 100 milliseconds, the syntax is
```
PAUSE TIME, 100
```

The macro execution will continue after the specified time interval has passed. No status message is displayed in this case. Note that the system timer has a resolution of about 10 milliseconds, so the actual delay introduced by PAUSE TIME will be between the requested time and the requested time plus (approximately) 10 milliseconds. The maximum allowed timer interval is 1,000,000 milliseconds (about 16.7 minutes).
If the keyword after the PAUSE command is THREADS, the macro execution will pause until all open windows complete whatever computation they are currently executing. This is a very useful feature when calling UPDATE ALL or LOADLENS using session files.

PIXEL

Turns on a single pixel on the current graphics screen.

Syntax:
```
PIXEL xcoord, ycoord
```

Discussion:
This feature is useful for making spot diagrams. See "GRAPHICS".

PLOT

The PLOT keyword supports a number of arguments used to simplify the task of creating plots of numeric data.

Syntax:
```
PLOT NEW
PLOT TITLE, string
PLOT TITLEX, string
PLOT TITLEY, string
PLOT BANNER, string
PLOT WINASPECT, type
PLOT COMM1, string
PLOT COMM2, string
PLOT COMM3, string
PLOT COMM4, string
PLOT COMM5, string
PLOT COMM6, string
PLOT RANGEX, minx, maxx
PLOT RANGEY, miny, maxy
PLOT CHECK, x_increment, y_increment
PLOT TICK, x_increment, y_increment
PLOT FORMATX, format_string
PLOT FORMATY, format_string
PLOT DATA, x_array, y_array, number_of_points, color, style, options
PLOT LINE, x1, y1, x2, y2
PLOT LABEL, x, y, angle, size, string
PLOT GO
```

Discussion:

PLOT NEW is used to initialize a new plot. All data related to any previously generated plot is discarded. This should always be the first PLOT command used when making a new graphic.

PLOT TITLE, PLOT TITLEX, and PLOT TITLEY are used to define the main plot title, the x-axis title, and the y-axis title, respectively. The parameter "string" may be a literal string in quotes or may be a string variable.

PLOT BANNER is used to define the banner title. The parameter "string" may be a literal string in quotes or may be a string variable.

PLOT WINASPECT defines the aspect ratio format of the window that will display the window. The valid values for type are the integers 0 through 3, which yield aspect ratios of 4x3, 5x3, 3x4, and 3x5, respectively. If no WINASPECT command is issued a default aspect ratio is used.

PLOT COMMx is used to define up to 6 comments that appear on the bottom of the plot. The parameter "string" may be a literal string in quotes or may be a string variable.

PLOT RANGEX and PLOT RANGEY define the minimum and maximum values of the plot in the x- and y- directions respectively. If no RANGEX or RANGEY command is issued, a default range will be selected.

PLOT CHECK is used to define the size of the "X" marks used to mark data points in the PLOT DATA command. The units are dimensionless fractions relative to the width of the display. If no CHECK command is issued, a default size will be selected.

PLOT TICK is used to define the increments between tick marks on the x- and y-axis, respectively. If no TICK command is issued, a default increment will be selected.

PLOT FORMATX and PLOT FORMATY are used to define the format strings for numeric labels on the x- and y-axis, respectively. The parameter "format_string" may be a literal string in quotes or may be a string variable. The format_string must be a valid C-language format specifier. A common format string is "%M.nf" where M is the total number of spaces in the output string and n is the number of figures after the decimal point. For example, the value for pi will print as "3.1415" if the format string is "%6.4f". To define an exponential format, the format string is of the form "%M.nE" which would yield a total of M characters and n is the number of figures after the decimal point. The value for pi would print as "3.14E+000" if the format_string was "%9.3E". If no FORMATX or FORMATY is specified a default format is used.

PLOT DATA is used to define a series of data points to be plotted. The variables x_array and y_array MUST be array variables of 1 dimension. For information on defining array variables see "Array variables". The number_of_points defines how many points in the array are to be used. The color argument is 0 for black, or 1-16 to select different pen colors. To define pen colors, see "Colors 1-12, Colors 13-24". The style argument defines the line style; the current supported values are 0 for a solid line, or 1-4 for various styles of dashed lines. The options value is 0 for no marks indicating the location of data points, 1 for both a line and data point marks, or 2 for no line and data point marks only. Multiple PLOT DATA commands may be issued, and each will create a separate line or curve to be plotted.

PLOT LINE makes a line between the points (x1, y1) and (x2, y2). The units of x and y here are normalized screen coordinates between 0.0 and 1.0. A maximum of 1000 lines may be defined.

PLOT LABEL prints any text string on the plot starting at the coordinates (x, y). The units of x and y here are normalized screen coordinates between 0.0 and 1.0. The angle is the angle from the +x direction in degrees. The size is a scaling constant greater than 0.0 that defines the size of the font. A size of 1.0 would print at the normal font size, while a value of 1.5 would print at 50% larger than normal size. The parameter "string" may be a literal string in quotes or may be a string variable. A maximum of 100 labels may be defined.

PLOT GO uses all settings and data from previous PLOT commands and generates the desired plot, and displays the plot in a window. After the plot is generated all the PLOT data is reset as though PLOT NEW had been executed.

Related Keywords:
PLOT2D

PLOT2D

The PLOT2D keyword supports a number of arguments used to simplify the task of creating plots of numeric data. This keyword makes it easy to generate surface, contour, grey scale, and false color plots.

Syntax:
```
PLOT2D NEW
PLOT2D TITLE, string
PLOT2D COMM1, string
PLOT2D COMM2, string
PLOT2D COMM3, string
PLOT2D COMM4, string
PLOT2D COMM5, string
```

```
PLOT2D RANGE, min, max
PLOT2D ASPECT, ratio
PLOT2D WINASPECT, type
PLOT2D DATA, arrayname
PLOT2D ACTIVECURSOR, left, right, bottom, top
PLOT2D DISPLAYTYPE, type
PLOT2D CONTOURINTERVAL, string
PLOT2D SURFACESCALE, scale
PLOT2D LOGPLOT, peak, decades
PLOT2D HIDEADDRESS, flag
PLOT2D CONFIG, configuration
PLOT2D GO
```

Discussion:

PLOT2D NEW is used to initialize a new plot. All data related to any previously generated plot2d is discarded. This should always be the first PLOT2D command used when making a new graphic.

PLOT2D TITLE is used to define the main plot title. The parameter "string" may be a literal string in quotes or may be a string variable.

PLOT2D COMMx is used to define up to 5 comments that appear on the bottom of the plot. The parameter "string" may be a literal string in quotes or may be a string variable.

PLOT2D RANGE define the minimum and maximum values of the plot. If no RANGE command is issued, a default range will be selected. The range should span the data values, or the data will be truncated to fit the range. The surface and contour plots ignore this command.

PLOT2D ASPECT defines the ratio of the x-width of the plot as compared to the y-width. The surface plots ignore this command.

PLOT2D WINASPECT defines the aspect ratio format of the window that will display the window. The valid values for type are the integers 0 through 3, which yield aspect ratios of 4x3, 5x3, 3x4, and 3x5, respectively. If no WINASPECT command is issued a default aspect ratio is used.

PLOT2D DATA is used to define the array variable containing the data to be plotted. The variable arrayname MUST be an array variables of 2 dimensions. For information on defining array variables see "Array variables". The number of points in the x and y directions is defined by the array dimensions. Any number of points is allowed, however, the minimum number of points in either direction is 5. The data will be assuming the two dimensions form a rectangular grid with uniform spacing along each dimension, although an overall aspect ratio for the plot may be defined using PLOT2D ASPECT. Only one PLOT2D DATA command may be issued.

PLOT2D ACTIVECURSOR is used to define the left, right, bottom, and top boundaries of the plotted data for display on contour, grey scale, and false color plots.

PLOT2D DISPLAYTYPE is used to select the type of plot generated. The value for type should be an integer between 1 and 6, inclusive, for surface, contour, grey scale, inverse grey scale, false color, and inverse false color, respectively.

PLOT2D CONTOURINTERVAL is used to define the contour format string. Only contour plots use this command. For details on the contour interval format string, see "The Contour Format String".

PLOT2D SURFACESCALE is used to define an overall vertical scaling of the surface plot. Only surface plots use this command. The default scaling value is 0.5 arbitrary units.

PLOT2D LOGPLOT is used to generate a log scale display. Only grey scale, inverse grey scale, false color, and inverse false color plots use this command. The peak value is an integer indicating the maximum power of 10 to plot. For example, is peak is 3, then the maximum value plotted will be 1E+03. The decades value is an integer indicating the number of log decades below the peak to plot. If peak is 3, and decades is 5, then the plot will span the range from 1.0E+03 to 1.0E-02. Note that the use of LOGPLOT does not actually take the logarithm of the input array data. The macro itself must take the log base 10 of the values in the array before calling PLOT2D GO.

PLOT2D HIDEADDRESS is used to hide or show the address data. If the flag value is zero, the address is shown, otherwise the address is not shown. See also "Address".

PLOT2D CONFIG is used to define the configuration number shown if the address box is displayed and the address box displays information about the current configuration. If configuration is 0, the address box will show the current configuration number. If the configuration is an integer between 1 and the number of configurations, inclusive, the specific configuration number will be shown. If configuration is an integer less than 0, then "All" is shown for the configuration number.

PLOT2D GO uses all settings and data from previous PLOT2D commands and generates the desired plot, and displays the plot in a window. After the plot is generated all the PLOT2D data is reset as though PLOT2D NEW had been executed.
Related Keywords:
PLOT

POLDEFINE

Defines the input polarization state for subsequent POLTRACE calls.
Syntax:
```
POLDEFINE Jx, Jy, PhaX, PhaY
```

Discussion:
The POLDEFINE keyword is used to define the input polarization state for subsequent polarization ray tracing. POLDEFINE requires the Jx and Jy electric field magnitudes, as well as the X and Y phase angles in degrees. For more information on the meaning of Jx and Jy, see "Defining the initial polarization". The input values are automatically normalized to have unity magnitude. The default values are 0, 1, 0, and 0, respectively. Once the polarization state is defined, it remains the same until changed.
Example:
```
POLDEFINE 2.0, 2.0, 45.0, -66.0
```

Related Keywords:
POLTRACE

POLTRACE

Calls the Zemax polarization ray tracing routines to trace a particular ray through the current system.
Syntax:
```
POLTRACE Hx, Hy, Px, Py, wavelength, vec, surf
```

Discussion:
The expressions Hx and Hy must evaluate to values between -1 and 1, and represent the normalized object coordinates. The pupil coordinates are specified by the expressions Px and Py, which also must be between -1 and 1. For more information about normalized coordinates, see the chapter "Conventions and Definitions" under "Normalized field and pupil coordinates". The wavelength expression must evaluate to an integer between 1 and the maximum number of defined wavelengths. The vec expression must evaluate to a number between 1 and 4, inclusive. The surf expression must evaluate to an integer between 1 and the number of surfaces, inclusive.
The input polarization state of the ray is defined by the POLDEFINE keyword.
Once the ray is traced, the polarization data for the ray is placed in the vector variable specified by the vec expression. For example, if the command "POLTRACE Hx, Hy, Px, Py, w, 2, n" is issued, the data will be stored in VEC2. The data is stored in the following format, where the first number in each line refers to the array position:
```
 0: n, the number of data entries in the vector
 1: The ray intensity after the surface
 2: E-Field X component, real
 3: E-Field Y component, real
 4: E-Field Z component, real
 5: E-Field X component, imaginary
 6: E-Field Y component, imaginary
 7: E-Field Z component, imaginary
 8: S-Polarization field amplitude reflection, real
 9: S-Polarization field amplitude reflection, imaginary
10: S-Polarization field amplitude transmission, real
11: S-Polarization field amplitude transmission, imaginary
12: P-Polarization field amplitude reflection, real
13: P-Polarization field amplitude reflection, imaginary
14: P-Polarization field amplitude transmission, real
15: P-Polarization field amplitude transmission, imaginary
16: E-Field X direction phase Px
17: E-Field Y direction phase Py
18: E-Field Z direction phase Pz
```

```
19: Major axis length of polarization ellipse
20: Minor axis length of polarization ellipse
21: Angle of polarization ellipse in radians
22: The surface number at which the ray was vignetted or zero if not vignetted
23: S-Polarization ray amplitude reflection, real
24: S-Polarization ray amplitude reflection, imaginary
25: S-Polarization ray amplitude transmission, real
26: S-Polarization ray amplitude transmission, imaginary
27: P-Polarization ray amplitude reflection, real
28: P-Polarization ray amplitude reflection, imaginary
29: P-Polarization ray amplitude transmission, real
30: P-Polarization ray amplitude transmission, imaginary
```

If the value in array position 0 is 0, then an error occurred and the polarization data is invalid. This may occur if the specified ray cannot be traced. See the RAYTRACE command to extract extended error information.
Example:
```
POLDEFINE 0, 1, 0, 0
POLTRACE 0, 1, 0, 0, pwav(), 1, nsur()
PRINT "Transmission of chief ray at primary wavelength is ", vec1(1)
```

Related Keywords:
POLDEFINE, RAYTRACE

POP
Computes the Physical Optics Propagation (POP) of a beam through the optical system and saves the surface by surface results to ZBF files. For a description of the POP feature see "Physical Optics Propagation". For information on the ZBF file format, see "Zemax Beam File (ZBF) binary format".
Syntax:
```
POP outfilename, lastsurface, settingsfilename
```

Discussion:
This keyword requires the name of the output ZBF file, an expression that evaluates to the last surface to propagate to, and optionally the name of a settings file. The filename must be enclosed in quotes if any blank or other special characters are used. The created ZBF files will be placed in the <pop> folder. No paths should be provided with the file names. The settings for the POP feature will be those settings previously saved for the current lens, unless a settings file name is provided. The settings file name must include the full path, name, and extension. To make adjustments to the settings, open a POP window, choose the appropriate settings, then press "Save". By default, all subsequent calls to POP within ZPL will use the saved settings. The exceptions are the output file name, which is specified as the first argument after the POP keyword, and the last surface number, which is optionally specified as the second argument after the POP keyword.
Example:
```
POP "pop_output.ZBF", 12, "MyPOP.CFG"
```

PRINT
Print is used to output constant text and variable data to either the screen or a file, depending upon the current status of the keyword OUTPUT.
Syntax:
```
PRINT
PRINT X
PRINT "The value of x is ", x
PRINT " x = ", x, " x + y = ", x + y
```

Discussion:
PRINT alone will print a blank line. PRINT with a list of text arguments and expressions will print each text string (enclosed in double quotes) and the numeric value of each expression. PRINT uses the numerical output format specified by FORMAT. If the last item in the list is followed by a comma, PRINT will not end the line with a carriage return.
Example:
```
X = 3
PRINT "X equals ",x
```

Related Keywords:
REWIND

PRINTFILE

Prints a text file.
Syntax:
```
PRINTFILE filename
```

Discussion:
The filename must be a valid file name. The file must be a text file (as would be created by OUTPUT and PRINT commands in ZPL) and must be in the current folder. PRINTFILE also closes the file if no CLOSE command has been executed.
Example:
```
OUTPUT "test.txt"
PRINT "Print this to the printer."
PRINTFILE "test.txt"
```

Related Keywords:
OPEN, OUTPUT, CLOSE, PRINT, PRINTFILE

PRINTWINDOW

Prints any open graphic or text window.
Syntax:
```
PRINTWINDOW winnum
```

Discussion:
The winnum value may be either an integer or an expression that evaluates to an integer. The integer winnum corresponds to the window number that should be printed. Zemax numbers windows sequentially as they are opened, starting with 1. Any closed windows are deleted from the window list, without renumbering the windows which remain. Any windows opened after another window has been closed will use the lowest window number available.
Example:
```
PRINTWINDOW 5
```

PWAV

Sets the primary wavelength.
Syntax:
```
PWAV n
```

Discussion:
The expression n is evaluated and the primary wavelength is set to the specified number. The UPDATE command must be issued before the new data takes effect.
Example:
```
PWAV 1
```

Related Functions:
WAVL, WWGT, PWAV

QUICKFOCUS

Adjusts the thickness of the surface prior to the image surface to minimize the specified criteria.
Syntax:
```
QUICKFOCUS mode, centroid
```

Discussion:

The expression for mode should evaluate to 0, 1, 2, or 3 for RMS spot radius, spot x, spot y, or wavefront OPD. The expression for centroid should evaluate to 0 or 1 to indicate the RMS should be referenced to the chief ray or image centroid, respectively. The "best" focus is chosen as a wavelength weighted average over all fields.
Example:
```
! Focus at best RMS wavefront to centroid
QUICKFOCUS 3, 1
```

RADI

This command is obsolete. See "SETSURFACEPROPERTY, SURP".

RANDOMIZE

RANDOMIZE seeds the random number generator.
Syntax:
```
RANDOMIZE seed
```

Discussion:
If seed evaluates to zero, Zemax seeds the random number generator with a value based upon the CPU clock. Otherwise, the value provided is used to seed the random number generator. Using the same seed will reproduce the identical series of random numbers created by the RAND function.
Example:
```
RANDOMIZE
RANDOMIZE 250
```

Related Functions:
RAND

RAYGENERATOR

Runs the ray generator, which is used to create a source ray file from a Radiant Source Model™ file.
Syntax:
```
RAYGENERATOR rsmx_file, ray_file, num_rays, min_azimuth, max_azimuth, min_inclination,
max_inclination, origination_method, origination_radius, spectrum_type, Arg1, Arg2, Arg3
```

Discussion:
RAYGENERATOR creates a spectral color format (.SDF) ray file (see "Source File") from a Radiant Source Model specified by the rsmx_file input argument. The output name for the ray file is given by the ray_file input argument. Both file names should be in quotes with the proper extension, and exclude paths. The RSMX file must be located in the <data>\Sources\Radiant Source Model Files folder. The output ray file will be placed in the <data>\Sources\Source Files folder.
The output ray file will have a number of rays defined by num_rays distributed over an angular range defined by min_azimuth, max_azimuth, min_inclination, and max_inclination. The azimuthal angle should be between 0 and 360 degrees. The inclination angle should be between 0 and 180 degrees. The ray origination method is a number between 0 and 11 corresponding to the 12 options available in the drop-down menu of the ray generator. Normal Plane is value 0, Closest to X Axis is value 1, etc. See the list in the ray generator dialog ("Generate Radiant Source Model Rays") for the complete set of supported options.
The origination_radius, which is always defined in millimeters, will only be used when the origination_method is 4 (On Sphere), 5 (On X Cylinder), 6 (On Y Cylinder), 7 (On Z Cylinder) or 11 (On Box). When the origination_method is 4, 5, 6, or 7, the origination_radius defines the radius of the sphere or cylinder from which rays are originated. When the origination_method is 11, rays originate from a cube centered on the source origin, with the origination_radius defining the half-width of the cube. Options to define end caps for the origination cylinder or an offset rectangular volume for the origination box are provided when using the ray generator dialog to create a ray file, but are not currently provided for the RAYGENERATOR keyword. Contact Zemax technical support if you require any options provided with the ray generator dialog to be added to the RAYGENERATOR keyword.
The spectrum type is defined by spec_type as follows: Measured = 0, Spectrum File = 1, Blackbody = 2, and Narrow Band = 3. The remaining arguments depend on the spectrum type. If a spectrum type doesn't use an argument, simply omit it. The measured spectrum type doesn't use any additional arguments. The spectrum file spectrum type uses Arg1 to

define the input spectrum file (*.SPCD). The file name should be in quotes with the extension and without a path specified. The file must be located in <data>\Sources\Spectrum Files. The blackbody spectrum type uses Arg1 to define the blackbody temperature in Kelvin, Arg2 to define the minimum wavelength, and Arg3 to define the maximum wavelength. The narrow band spectrum type uses Arg1 to define the central wavelength and Arg2 to define the spectral width.

RAYTRACE

Calls the Zemax ray tracing routines to trace a particular ray through the current system.
Syntax:
```
RAYTRACE hx, hy, px, py, wavelength
```

Discussion:
The expressions hx and hy must evaluate to values between -1 and 1, and represent the normalized field coordinates. The pupil coordinates are specified by the expressions px and py, which also must be between -1 and 1. For more information about normalized coordinates see "Normalized field coordinates". The wavelength expression is optional, defaulting to the primary wavelength, but if supplied must evaluate to an integer between 1 and the maximum number of defined wavelengths.
Once the ray is traced, the ray intercept coordinates and direction cosines may be determined using the ZPL functions RAYX, RAYY, RAYZ, RAYL, RAYM, and RAYN (use RAGX, RAGY, RAGZ, RAGL, RAGM, and RAGN to get results in global coordinates). If an error occurred during ray tracing, the function RAYE (for RAY Error) will return a value other than zero. If RAYE is negative, it indicates that total internal reflection occurred at the surface whose number is the absolute value of the value returned. If RAYE is greater than zero, then the ray missed the surface number returned. Checking RAYE is optional, however, the RAYX, RAYY, ... functions may return invalid data if RAYE is not zero. The functions RANX, RANY, and RANZ return the intercept surface normal direction cosines, and OPDC returns the optical path difference for the ray. The function RAYV returns the surface number at which the ray was vignetted, or it returns zero if the ray was not vignetted. Values returned for surfaces past the surface of vignetting may not be accurate.
Example:
```
PRINT "Tracing the marginal ray at primary wavelength!"
n = NSUR()
RAYTRACE 0,0,0,1
y = RAYY(n)
PRINT "The ray intercept is ", y
PRINT "Tracing the chief ray at maximum wavelength!"
RAYTRACE 0,1,0,0,NWAV()
y = RAYY(n)
PRINT "The ray intercept is ", y
```

Related Keywords:
RAYTRACEX

RAYTRACEX

Calls the Zemax ray tracing routines to trace a particular ray from any starting surface through the current system.
Syntax:
```
RAYTRACEX x, y, z, l, m, n, surf, wavelength
```

Discussion:
The expressions x, y, z, l, m, and n define the input ray position and direction cosines in the local coordinates of the starting surface. The surface expression must evaluate to an integer between 0 and the number of surfaces minus one, inclusive. The wavelength expression is optional, defaulting to the primary wavelength, but if supplied must evaluate to an integer between 1 and the maximum number of defined wavelengths.
If the object has a thickness of infinity, and the surf parameter is zero, then the input coordinates are assumed to be relative to the first surface rather than the object surface; although the ray will still be defined in object space media. Otherwise, Zemax uses the specified coordinates without alteration.
Once the ray is traced, the ray intercept coordinates and direction cosines may be determined using the ZPL functions RAYX, RAYY, RAYZ, RAYL, RAYM, and RAYN (use RAGX, RAGY, RAGZ, RAGL, RAGM, and RAGN to get results in global coordinates). Note only data from surface AFTER the "surf" surface will be valid.

If an error occurred during ray tracing, the function RAYE (for RAY Error) will return a value other than zero. If RAYE is negative, it indicates that total internal reflection occurred at the surface whose number is the absolute value of the value returned. If RAYE is greater than zero, then the ray missed the surface number returned. Checking RAYE is optional, however, the RAYX, RAYY, ... functions may return invalid data if RAYE is not zero. The functions RANX, RANY, and RANZ return the intercept surface normal direction cosines, and RAYT returns the optical path length up to the surface for the ray. The function RAYV returns the surface number at which the ray was vignetted, or it returns zero if the ray was not vignetted. Values returned for surfaces past the surface of vignetting may not be accurate.

Example:
```
n = NSUR()
RAYTRACEX 0,1,0,0,0,1,0,NWAV()
y = RAYY(n)
PRINT "The ray intercept is ", y
```

Related Keywords:
RAYTRACE

READ

Reads data from an existing text file opened for reading by the OPEN command.

Syntax:
```
READ x
READ x, y
READ x,y,z,a,b,c,q
```

Discussion:
The file must be already open, see the keyword OPEN for details. Each READ command reads a single line from the file. The first valid data field from this line is placed in the variable first listed. The data from the second field is placed in the second variable listed, if any. Therefore, the number of variables listed in the read command should match the number of columns in the text file. Numeric data in the file should be delimited by spaces. The data may be in free-form, and is internally promoted to double precision. A maximum of 2000 characters can be read in on any single line. The maximum number of variable arguments is 199; for reading longer lines with more arguments use READNEXT instead. The variables listed must be valid ZPL variable names. READNEXT does not support array variable names as arguments. The workaround is to read the data into a scalar variable and then set the array variable to the scalar variable value in a subsequent line like this:
```
READ x
data(i, j) = x
```

Always CLOSE a file after all the data has been read. See the function EOFF.

Example:
```
PRINT "Reading the double-column file TEST.DAT!"
OPEN "C:\DATA\TEST.DAT"
READ x1, y1
READ x2, y2
READ x3, y3
CLOSE
```

Related Functions:
EOFF
Related Keywords:
OPEN, CLOSE, READNEXT, READSKIP, READSTRING

READNEXT

Reads data from an existing text file opened for reading by the OPEN command.

Syntax:
```
READNEXT x
READNEXT x, y
READNEXT x,y,z,a,b,c,q
```

Discussion:

READNEXT is almost identical to READ. The key difference is READ will read the entire data line from the opened file, up to the newline character, while READNEXT reads only enough characters to fill the number of arguments. For example, if a data file contains a line with this data:

```
3.0 4.0 5.0
```

The following two READNEXT commands will read the values 3.0, 4.0, and 5.0 for x, y, and z:

```
READNEXT x, y
READNEXT z
```

READNEXT is more useful than READ if the line is very long, or the number of arguments is large. READNEXT does not support array variable names as arguments. The workaround is to read the data into a scalar variable and then set the array variable to the scalar variable value in a subsequent line like this:

```
READNEXT x
data(i, j) = x
```

Example:

```
OPEN "C:\DATA\TEST.DAT"
READNEXT x1, x2
READNEXT x3
CLOSE
```

Related Keywords:
OPEN, CLOSE, READ, READSKIP, READSTRING

READSKIP

Reads any number of unwanted characters from an existing text file opened for reading by the OPEN command.
Syntax:

```
READSKIP n
```

Discussion:
The file must be already open, see the keyword OPEN for details. If n is positive, READSKIP reads off exactly n characters from the file. If n is negative, READSKIP reads to the end of the current line. The read characters are discarded. This function allows faster access to large files by effectively skipping over any number of unwanted characters. Always CLOSE a file after all the data has been read. See the function EOFF.
Example:

```
PRINT "Reading the contents of file TEST.DAT!"
OPEN TEST.DAT
READSKIP 59
READSTRING A$
PRINT A$
CLOSE
```

Related Keywords:
OPEN, CLOSE READ, READNEXT, READSTRING

READSTRING

Reads data from an existing text file opened for reading by the OPEN command.
Syntax:

```
READSTRING A$
```

Discussion:
The file must be already open, see the keyword OPEN for details. Each READ command reads a single line from the file. The entire line read is placed in the variable listed. A maximum of 359 characters can be read in and placed in a single string. The variable listed must be a valid ZPL string variable name, although it does not need to be previously referenced. Always CLOSE a file after all the data has been read. See the function EOFF.
Example:

```
PRINT "Reading the contents of file TEST.DAT!"
OPEN TEST.DAT
```

```
READSTRING A$
PRINT A$
CLOSE
```

Related Keywords:
OPEN, CLOSE READ, READNEXT, READSKIP

RELEASE
See "Array variables".

RELOADOBJECTS
Reloads NSC objects into the NSC Editor.
Syntax:
```
RELOADOBJECTS surface, object
```

Discussion:
The expression for surface must evaluate to an integer surface number corresponding to the non-sequential component surface. For NSC mode, use 1. The expression for object must evaluate to an integer object number, or zero to reload all objects. This keyword may take a significant amount of time to execute, depending upon the number and type of objects defined.
Example:
```
RELOADOBJECTS 1, 0
```

REM,!,#
REM, !, and # are used to define remarks or comments.
Syntax:
```
REM (text)
! (text)
(valid zpl line) # (text)
```

Discussion:
The exclamation symbol may also be used to indicate a remark. Both the REM command and the "!" symbol are only recognized as remark indicators if they appear at the very beginning of the line. The # symbol may be used anywhere on the line, but will only be interpreted as the beginning of a comment if the # is not inside a string. Everything after the # symbol is ignored.
Example:
```
REM any text can be placed after the REM command.
! any text can also be placed
! after the exclamation symbol.
x = 5 # this syntax allows comments to be placed on the same line as a command.
```

REMOVEVARIABLES
Sets all currently defined variables to fixed status.
Syntax:
```
REMOVEVARIABLES
```

RESUMEUPDATES

SUSPENDUPDATES prevents any UI editor windows from being updated by ZPL commands while suspended, and RESUMEUPDATES ends the suspension. See SUSPENDUPDATES for more information.

RENAMEFILE
 RENAMEFILE is used to rename a file.
Syntax:

```
      RENAMEFILE oldfilename, newfilename
```

Discussion:
This keyword requires two file names, defined as literal string expressions in quotes or as string variables. The file oldfilename is renamed newfilename.

Example:
```
  RENAMEFILE AFILE$, BFILE$
```

Related Keywords:
```
  COPYFILE
  DELETEFILE
```

RETURN

See GOSUB.

REWIND

 REWIND erases the last line printed by the PRINT command, up to the previous end of line. This allows printing a counter or other data over an existing line in the text output file.

Syntax:
```
  REWIND
```

Example:
```
  PRINT "First line"
  REWIND
  PRINT "New First line"
```

Related Keywords:
PRINT

SAVEARCHIVE

Saves the current lens to a Zemax archive (*.ZAR) file.

Syntax:
```
  SAVEARCHIVE filename
```

Discussion:
The filename must include the name of the archive file including the file extension. If the filename does not include a complete path to the archive file, the default folder for lenses is assumed. See "Backup To Archive File".

SAVEDETECTOR

Saves the data currently on an NSC Detector Rectangle, Detector Color, Detector Polar, or Detector Volume object to a file.

Syntax:
```
  SAVEDETECTOR surf, object, filename
```

Discussion:
This keyword requires numeric expressions that specify the surface and object, and a file name with or without a full path. Surf is the surface number of the non-sequential group; use 1 when using non-sequential mode. Object is the object number of the detector object. The filename may include the full path, if no path is provided the path of the current lens file is used. The extension should be DDR, DDC, DDP, or DDV for Detector Rectangle, Color, Polar, and Volume objects, respectively.

Related Keywords:
LOADDETECTOR

SAVELENS

Saves the current lens file.
Syntax:

```
SAVELENS filename, session
```

Discussion:

SAVELENS will save the current lens file to the specified file name. The name of the current lens in memory will also be changed. If the file name is absent, then the lens data is stored in the current file name. If the session argument evaluates to anything other than zero, the session file will also be saved.

Example:
```
SAVELENS
SAVELENS "NEWCOPY.ZMX"
SAVELENS NEW$
```

Related Keywords:
LOADLENS

SAVEMERIT

Saves the current merit function to a file.
Syntax:
```
SAVEMERIT filename
```

Discussion:

SAVEMERIT will save the current merit function to a file. If the filename contains the complete path, such as C:\MYDIR\MYLENS.MF, then the specified path will be used. If the path is left off, then the <data>\MeritFunction folder will be used (see "Folders"). See also LOADMERIT.

SAVETOLERANCE

Saves the current tolerances to a file.
Syntax:
```
SAVETOLERANCE "filename"
SAVETOLERANCE file$
```

Discussion:

If the filename contains the complete path, such as C:\MYDIR\MYLENS.TOL, then the specified file will be created. If the path is left off, then the <data>\Tolerance folder will be used (see "Folders"). See also LOADTOLERANCE.

SAVEWINDOW

Saves the text from any text window to a file.
Syntax:
```
SAVEWINDOW winnum, filename
```

Discussion:

The winnum value may be either an integer or an expression that evaluates to an integer. The integer winnum corresponds to the text window number that should be saved to a file. Zemax numbers windows sequentially as they are opened, starting with 1. Any closed windows are deleted from the window list, without renumbering the windows which remain. Any windows opened after another window has been closed will use the lowest window number available.
Example:
```
SAVEWINDOW 1, "C:\TEMP\TEXTFILE.TXT"
SAVEWINDOW 3, A$
```

SCATTER

Used to control whether sequential surface scattering is done while tracing rays.
Syntax:
```
SCATTER ON
SCATTER OFF
```

Discussion:

The default condition at the start of a macro is SCATTER OFF; and all rays will be traced deterministically. If SCATTER ON is executed, then sequential surface scattering will be enabled for all subsequent RAYTRACE commands.

SDIA

This command is obsolete. See "SETSURFACEPROPERTY, SURP".

SETAIM

Sets the state of the ray aiming function.
Syntax:
```
SETAIM state
```

Discussion:
This keyword requires one numeric expression that must evaluate to either 0 or 1. The state is a code which is 0 for ray aiming off and 1 for ray aiming on.
Example:
```
SETAIM 1
```

Related Keywords:
SETAIMDATA

SETAIMDATA

Sets various data for the ray aiming function.
Syntax:
```
SETAIMDATA code, value
```

Discussion:
The code values are used as follows:

Code	Property
1	Sets "Use Ray Aiming Cache" to true if value is 1, or false if value is 0.
2	Sets "Robust Ray Aiming" to true if value is 1, or false if value is 0.
3	Sets "Scale Pupil Shift Factors by Field" to true if value is 1, or false if value is 0.
4, 5, 6	Sets the value of the x, y, and z pupil shift, respectively.
7, 8	Sets the value of the x and y pupil compress, respectively.

Example:
```
SETAIMDATA 5, 0.34
```

Related Keywords:
SETAIM

SETAPODIZATION

This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

SETCONFIG

Sets the current configuration for multi-configuration (zoom) systems.
Syntax:
```
SETCONFIG config
```

Discussion:
The expression config must evaluate to an integer between 1 and the maximum number of configurations.
Example:
```
SETCONFIG 4
```

Related Functions:
CONF, NCON

SETDETECTOR
 Sets the coherent or incoherent detector data for any pixel on a detector rectangle object.
Syntax:
```
SETDETECTOR surf, object, pixel, datatype, value
```

Discussion:
This keyword requires numeric expressions that specify the surface, object, pixel, and data type (all integers) and the new detector value. Surf is the surface number of the non-sequential group; use 1 when using non- sequential mode. Object is the object number of the detector. The object must be a detector rectangle. Pixel is a number between 1 and the number of pixels the object supports. Datatype should be 0 for incoherent intensity, 1 for incoherent intensity in angle space, 2 for coherent real part, 3 for coherent imaginary part, and 4 for coherent amplitude. For more information on the use and meaning of these values see "Detector Rectangle object". The units for incoherent intensity are source units (see "Source Units"). The units for coherent amplitude are the square root of source units.
Example:
```
SETDETECTOR 1, 5, 12, 0, 1.0056
```

Related Functions:
NSDD

SETMCOPERAND
 Sets any row or configuration of the Multi-Configuration Editor to any numeric value.
Syntax:
```
SETMCOPERAND row, config, value, datatype
```

Discussion:
This keyword requires numeric expressions that evaluate to integers specifying the row and configuration of the Multi-Configuration Editor.
If the config number is 0, then the value is interpreted as follows:
datatype = 0, value is a string literal or variable that specifies the name of the operand.
datatype = 1, 2, or 3, value is the number 1, 2, or 3 value used as part of the multi-configuration operand definition. See the description of the multi-configuration operand numbers defined in "Summary of multi- configuration operands".
If the config number corresponds to a defined configuration then the value is interpreted as follows:
datatype = 0, value is the value of the operand.
datatype = 1, value is the pickup offset of the operand.
datatype = 2, value is the pickup scale of the operand.
datatype = 3, value is the status of the operand, 0 for fixed, 1 for variable, 2 for pickup, 3 for thermal pickup.
datatype = 4, value is the pickup configuration number.
datatype = 5, value is the pickup row number.
Example:
```
SETMCOPERAND 3, 4, somevalue, 0
SETMCOPERAND 1, 0, "THIC", 0
```

Related Functions:
MCOP

SETNSCPARAMETER
Sets the parameter values of any object in the NSC editor.
Syntax:

```
SETNSCPARAMETER surface, object, parameter, value
```

Discussion:
This keyword requires 3 numeric expressions that evaluate to integers specifying the non-sequential components surface number, the object number, and the parameter number. The fourth argument is the new value for the specified parameter.
Example:
```
SETNSCPARAMETER 4, 2, 15, newp15value
```

Related Functions:
NPOS, NPAR
Related Keywords:
INSERTOBJECT, SETNSCPOSITION, SETNSCPROPERTY

SETNSCPOSITION
Sets the x, y, z or tilt x, tilt y, tilt z position of any object in the NSC editor.
Syntax:
```
SETNSCPOSITION surface, object, code, value
```

Discussion:
This keyword requires 3 numeric expressions that evaluate to integers specifying the non-sequential components surface number, the object number, and a code. The code is 1 through 6 for x, y, z, tilt x, tilt y, tilt z, respectively. The fourth argument is the new value for the specified position.
Example:
```
SETNSCPOSITION 4, 2, 2, newyvalue
```

Related Functions:
NPOS, NPAR
Related Keywords:
INSERTOBJECT, SETNSCPARAMETER, SETNSCPROPERTY

SETNSCPROPERTY
Sets properties of NSC objects.
Syntax:
```
SETNSCPROPERTY surface, object, code, face, value
```

Discussion:
This keyword requires 4 numeric expressions that evaluate to integers specifying the non-sequential components surface number (use 1 if the program mode is non-sequential), the object number, a code which specifies what property of the object is being modified, and the face number (use 0 if not applicable to the property being set). The fifth argument is the new value for the specified property, and it may be either text in quotes, a string variable, or a numeric expression. The code is as follows:

Code	Property
The following codes set values on the NSC Editor.	
1	Sets the object comment.
2	Sets the reference object number.
3	Sets the "inside of" object number.
4	Sets the object material.

	The following codes set values on the Type tab of the Object Properties dialog.
0	Sets the object type. The value should be the name of the object, such as "NSC_SLEN" for the standard lens. The names for each object type are listed in the Prescription Report for each object type in the NSC editor. All NSC object names start with "NSC_".
13	Sets User Defined Aperture, use 1 for checked, 0 for unchecked.
14	Sets the User Defined Aperture file name.
15	Sets the "Use Global XYZ Rotation Order" checkbox, use 1 for checked, 0 for unchecked.
16	Sets the "Rays Ignore This Object" combo box, use 0 for Never, 1 for Always, and 2 for On Launch.
17	Sets the "Object Is A Detector" checkbox, use 1 for checked, 0 for unchecked.
18	Sets the "Consider Objects" list. The argument should be a string listing the object numbers to consider delimited by spaces, such as "2 5 14".
19	Sets the "Ignore Objects" list. The argument should be a string listing the object numbers to ignore delimited by spaces, such as "1 3 7".
20	Sets the "Use Pixel Interpolation" checkbox, use 1 for checked, 0 for unchecked.
30	Sets the "Use Consider/Ignore Objects When Splitting" checkbox, use 1 for checked, 0 for unchecked.
	The following codes set values on the Coat/Scatter tab of the Object Properties dialog.
5	Sets the coating name for the specified face.
6	Sets the profile name for the specified face.
7	Sets the scatter mode for the specified face: 0 = none, 1 = Lambertian, 2 = Gaussian, 3 = ABg, 4 = User Defined, 5 = BSDF, 6 = ABg File, 7 = IS Scatter Catalog.
8	Sets the scatter fraction for the specified face.
9	Sets the number of scatter rays for the specified face.
10	Sets the Gaussian sigma (Gaussian scatter model) or the sample orientation angle (BSDF or IS Scatter Catalog scatter models) for the specified face.
11	Sets the reflect ABg data name for the specified face.

12	Sets the transmit ABg data name for the specified face.
27	Sets the name of the user defined scattering DLL.
21-26	Sets parameter values on the user defined scattering DLL.
28	Sets the name of the user defined scattering data file.
29	Sets the "Face Is" property for the specified face. Use 0 for "Object Default", 1 for "Reflective", and 2 for "Absorbing".
31	Sets the reflect BSDF data file for the specified face. The value should be the name of the BSDF file with no path (i.e. BrownVinyl.bsdf).
32	Sets the transmit BSDF data file for the specified face. The value should be the name of the BSDF file with no path (i.e. BrownVinyl.bsdf).
33	Sets the reflect ABg File data file for the specified face. The value should be the name of the ABGF file with no path (e.g. SampleABGF.abgf).
34	Sets the transmit ABg File data file for the specified face. The value should be the name of the ABGF file with no path (e.g. SampleABGF.abgf).
35	Sets the reflect IS Scatter Catalog data file for the specified face. The value should be the name of the ISX file with no path (e.g. BrownVinyl.ISX).
36	Sets the transmit IS Scatter Catalog data file for the specified face. The value should be the name of the ISX file with no path (e.g. BrownVinyl.ISX).
37	Sets the Thin Window Scattering option for the specified face. Use 0 to turn the option off (i.e. unchecked option in checkbox) and 1 to turn the option on (i.e. checked option in checkbox).
38	Sets the sample side R for IS Scatter Catalog scattering. Use 0 for front and 1 for back.
39	Sets the sample side T for IS Scatter Catalog scattering. Use 0 for front and 1 for back.
40	Sets the sampling R for IS Scatter Catalog scattering. Use 0 for 5 degrees, 1 for 2 degrees, and 2 for 1 degree.
41	Sets the sampling T for IS Scatter Catalog scattering. Use 0 for 5 degrees, 1 for 2 degrees, and

	2 for 1 degree.
The following codes set values on the Bulk Scattering tab of the Object Properties dialog.	
81	Sets the "Model" value on the bulk scattering tab. Use 0 for "No Bulk Scattering", 1 for "Angle Scattering", and 2 for "DLL Defined Scattering".
82	Sets the mean free path to use for bulk scattering.
83	Sets the angle to use for bulk scattering.
84	Sets the name of the DLL to use for bulk scattering.
85	Sets the parameter value to pass to the DLL, where the face value is used to specify which parameter is being defined. The first parameter is 1, the second is 2, etc.
86	Sets the wavelength shift string.
The following codes set values on the Diffraction tab of the Object Properties dialog.	
91	Sets the "Split" value on the diffraction tab. Use 0 for "Don't Split By Order", 1 for "Split By Table Below", and 2 for "Split By DLL Function".
92	Sets the name of the DLL to use for diffraction splitting.
93	Sets the Start Order value.
94	Sets the Stop Order value.
95, 96	Sets the parameter values on the diffraction tab. These are the parameters passed to the diffraction splitting DLL as well as the order efficiency values used by the "split by table below" option. The face value is used to specify which parameter is being defined. The first parameter is 1, the second is 2, etc. The code 95 is used for reflection properties, and 96 for transmission.
The following codes set values on the Sources tab of the Object Properties dialog.	
101	Sets the source object random polarization. Use 1 for checked, 0 for unchecked.
102	Sets the source object reverse rays option. Use 1 for checked, 0 for unchecked.
103	Sets the source object Jones X value.
104	Sets the source object Jones Y value.
105	Sets the source object Phase X value.

106	Sets the source object Phase Y value.
107	Sets the source object initial phase in degrees value.
108	Sets the source object coherence length value.
109	Sets the source object pre-propagation value.
110	Sets the source object sampling method; 0 for random, 1 for Sobol sampling.
111	Sets the source object bulk scatter method; 0 for many, 1 for once, 2 for never.
112	Sets the array mode; 0 for none, 1 for rectangular, 2 for circular, 3 for hexapolar, and 4 for hexagonal.
113	Sets the source color mode. For a complete list of the available modes, see "Defining the color and spectral content of sources". The source color modes are numbered starting with 0 for the System Wavelengths, and then from 1 through the last model listed in the dialog box control.
114-116	Sets the number of spectrum steps, start wavelength, and end wavelength, respectively.
117	Sets the name of the spectrum file.
161-162	Sets the array mode integer arguments 1 and 2.
165-166	Sets the array mode double precision arguments 1 and 2.
181-183	Sets the source color mode arguments, for example, the XYZ values of the Tristimulus.

The following codes set values on the Grin tab of the Object Properties dialog.

121	Sets the "Use DLL Defined Grin Media" checkbox. Use 1 for checked, 0 for unchecked.
122	Sets the Maximum Step Size value.
123	Sets the DLL name.
124	Sets the Grin DLL parameters. These are the parameters passed to the DLL. The face value is used to specify which parameter is being defined. The first parameter is 1, the second is 2, etc.

The following codes set values on the Draw tab of the Object Properties dialog.

141	Sets the do not draw object checkbox. Use 1 for checked, 0 for unchecked.

142	Sets the object opacity. Use 0 for 100%, 1 for 90%, 2 for 80%, etc.
143	Sets the drawing resolution for the object. Use 0 for Standard, 1 for Medium, 2 for High, 3 for Presentation, and 4 for Custom.
144	Sets the value of the first resolution input to the drawing resolution (e.g. "Angular" resolution for the Annular Aspheric Lens object) when the drawing resolution itself is set to Custom. The drawing resolution must first be set to Custom before this can be used.
145	Sets the value of the second resolution input to the drawing resolution (e.g. "Radial" resolution for the Annular Aspheric Lens object) when the drawing resolution itself is set to Custom. The drawing resolution must first be set to Custom before this can be used.

The following codes set values on the Scatter To tab of the Object Properties dialog.

151	Sets the scatter to method. Use 0 for scatter to list, and 1 for importance sampling.
152	Sets the Importance Sampling target data. The argument should be a string listing the ray number, the object number, the size, and the limit value, all separated by spaces Here is a sample syntax to set the Importance Sampling data for ray 3, object 6, size 3.5, and limit 0.6: "3 6 3.5 0.6".
153	Sets the "Scatter To List" values. The argument should be a string listing the object numbers to scatter to delimited by spaces, such as "4 6 19".

The following codes set values on the Birefringence tab of the Object Properties dialog.

171	Sets the Birefringent Media checkbox. Use 0 for unchecked, and 1 for checked.
172	Sets the Birefring ent Media Mode. Use 0 for Trace ordinary and extraordinary rays, 1 for Trace only ordinary rays, 2 for Trace only extraordinary rays, and 3 for Waveplate mode.
173	Sets the Birefringent Media Reflections status. Use 0 for Trace reflected and refracted rays, 1 for Trace only refracted rays, and 2 for Trace only reflected rays.
174-176	Sets the Ax, Ay, and Az values.
177	Sets the Axis Length.

The following codes do not set values, but are included here to return values for the function NPRO.

200	Used by function NPRO to determine the index of refraction of an object. The syntax is NPRO(surface, object, 200, wavenumber)
201-203	Used by function NPRO to determine the nd (201), vd (202), and dpgf (203) parameters of an object using a model glass. The syntax is

NPRO(surface, object, 201, 0)

Other code values will be added in the future to set other properties of NSC objects as requested by users.

Example:
```
SETNSCPROPERTY 1, 2, 0, 0, "NSC_SLEN"
```

Related Functions:
NPOS, NPAR, NPRO
Related Keywords:
INSERTOBJECT, SETNSCPARAMETER, SETNSCPOSITION

SETOPERAND

Sets any row or column of the Merit Function Editor to any numeric value.
Syntax:
```
SETOPERAND row, col, value
```

Discussion:
This keyword requires numeric expressions that evaluate to integers specifying the row and column of the Merit Function Editor. The col integer is 1 for operand type, 2 for int1, 3 for int2, 4-7 for data1-data4, 8 for target, and 9 for weight. To set the comment string associated with an operand use col 10. To set the operand type, use a col value of 1 and the integer operand returned by the function ONUM. An alternate method for setting the operand type is to use a col value of 11 with the string name of the operand type, such as EFFL or DIST. If col is 10 or 11, value should be a string constant or variable. The col integer is 12 for data5 and 13 for data6. Note the value and percent contribution columns cannot be set but must be computed.
Example:
```
SETOPERAND 1, 8, tarvalue
SETOPERAND 3, 11, "EFFL"
SETOPERAND 5, 10, "Operand Number 5"
```

Related Functions:
MFCN, OPER, ONUM

SETSTDD

This command is obsolete. See "SETSURFACEPROPERTY, SURP".

SETSURFACEPROPERTY, SURP

Sets properties of surfaces.
Syntax:
```
SETSURFACEPROPERTY surface, code, value1, value2
SURP surface, code, value1, value2
```

Discussion:
Surface is an expression that evaluates to an integer specifying the surface number. The code may either be an expression that evaluates to an integer or a mnemonic which specifies what property of the surface is being modified. The third and fourth arguments are the new values for the specified property, and they may be either text in quotes, a string variable, or a numeric expression, depending upon the code. For most codes, the property value being modified is defined by the value1 argument. A few operands require both a value1 and a value2, as described in the table below.
If the property being modified is under control of the Multi-Configuration Editor, then the multi-configuration data for the current configuration only will also be modified to reflect the changed property.
To set a surface as the stop surface, see "STOPSURF".
SURP is a shorthand for SETSURFACEPROPERTY and is functionally identical.

Code	Property

Basic Surface Data. See "Lens Data".	
0 or TYPE	Surface type. The value should be the name of the object, such as "STANDARD" for the standard surface. The names for each surface type are listed in the Prescription Report in the Surface Data Summary for each surface type currently in the Lens Data Editor.
	To change the surface type to a user defined surface, first set the DLL name using code 9 (SDLL), then set the new surface type to USERSURF.
	See also Code 17.
1 or COMM	Comment.
2 or CURV	Curvature (not radius) in inverse lens units. Use zero for an infinite radius.
3 or THIC	Thickness in lens units.
4 or GLAS	Glass name. See also Code 18.
5 or CONI	Conic constant.
6 or SDIA	Semi-diameter. If the value is zero or positive, the semi-diameter solve is set to "Fixed". If the value is negative, the semi-diameter solve is set to "Automatic" and the semi-diameter will be computed with the next UPDATE keyword.
7 or TCE	Thermal coefficient of expansion.
8 or COAT	Coating name. Use a blank string for value1 to remove the coating.
9 or SDLL	User defined surface DLL name.
10 or PARM	Parameter value. Value1 is the new value. Value2 is the parameter number.
11 or EDVA	Extra Data value. Value1 is the new value. Value2 is the extra data number.
12	Surface color, Use 0 for default.
13	Surface opacity.
14	Row color.
15	Surface cannot be hyperhemispheric. Use 1 to avoid surface being hyperhemispheric.
16	Ignore surface. Use 1 to ignore surface, 0 to not ignore surface.
17 or CODE	The integer code for the surface type. The integer code is an alternative to the surface name used by code 0. For full details see the documentation for

	code 0 above.
18 or GLAN	Glass number. See also Code 4.

Surface aperture data. See "Surface properties aperture tab".

20 or ATYP	Surface aperture type code.
21 or APP1	Surface aperture parameter 1.
22 or APP2	Surface aperture parameter 2.
23 or APDX	Surface aperture decenter x.
24 or APDY	Surface aperture decenter y.
25 or UDA	User Defined Aperture (UDA) file name.
26 or APPU	Surface aperture pick up from surface number. Use 0 for no pickup.

Physical Optics Propagation Settings. See "Surface specific settings".

30	Physical Optics setting "Use Rays To Propagate To Next Surface". Use 1 for true, 0 for false.
31	Physical Optics setting "Do Not Rescale Beam Size Using Ray Data". Use 1 for true, 0 for false.
32	Physical Optics setting "Use Angular Spectrum Propagator". Use 1 for true, 0 for false.
33	Physical Optics setting "Draw ZBF On Shaded Model". Use 1 for true, 0 for false.
34	Physical Optics setting "Recompute Pilot Beam Parameters". Use 1 for true, 0 for false.
35	Physical Optics setting "Resample After Refraction". Use 1 for true, 0 for false.
36	Physical Optics setting "Auto Resample". Use 1 for true, 0 for false.
37	Physical Optics setting "New X Sampling". Use 1 for 32, 2 for 64, etc.
38	Physical Optics setting "New Y Sampling". Use 1 for 32, 2 for 64, etc.
39	Physical Optics setting "New X-Width". New total x direction width of array.
40	Physical Optics setting "New Y-Width". New total y direction width of array.

41	Physical Optics setting "Output Pilot Radius". Use 0 for best fit, 1 for shorter, 2 for longer, 3 for x, 4 for y, 5 for plane, 6 for user.
42, 43	Physical Optics setting "X-Radius" and "Y-Radius", respectively.
44	Physical Optics setting "Use X-axis Reference". Use 1 for true, 0 for false.
Coating Settings. See "Surface coating tab". See also code 8 above.	
50	Use Layer Multipliers and Index Offsets. Use 1 for true, 0 for false.
51	Layer Multiplier value. Value1 is the new value. Value2 is the layer number.
52	Layer Multiplier status. Value 1 is the status, use 0 for fixed, 1 for variable, or n+1 for pickup from layer n. Value2 is the layer number.
53	Layer Index Offset value. Value1 is the new value. Value2 is the layer number.
54	Layer Index Offset status. Value 1 is the status, use 0 for fixed, 1 for variable, or n+1 for pickup from layer n. Value2 is the layer number.
55	Layer Extinction Offset value. Value1 is the new value. Value2 is the layer number.
56	Layer Extinction Offset status. Value 1 is the status, use 0 for fixed, 1 for variable, or n+1 for pickup from layer n. Value2 is the layer number.
Surface Tilt and Decenter Data. See "Surface tilt/decenter tab".	
60 or BOR	Before tilt and decenter order. Use 0 for dec/tilt, 1 for tilt/dec.
61 or BDX	Before decenter x.
62 or BDY	Before decenter y.
63 or BTX	Before tilt about x.
64 or BTY	Before tilt about y.
65 or BTZ	Before tilt about z.
66 or APU	After pick up status: 0 for explicit, 1/2 for pickup/reverse current surface, 3/4 for pickup/reverse current surface minus 1, 5/6 for pickup/reverse current surface minus 2, etc...
70 or AOR	After tilt and decenter order. Use 0 for dec/tilt, 1 for tilt/dec.
71 or ADX	After decenter x.

72 or ADY	After decenter y.
73 or ATX	After tilt about x.
74 or ATY	After tilt about y.
75 or ATZ	After tilt about z.
76	Coordinate Return status. Valid only on Coordinate Break surfaces. Use 0 for None, 1 for Orientation Only, 2 for Orientation XY, and 3 for Orientation XYZ.
77	Coordinate Return To Surface. Valid only on Coordinate Break surfaces.

Surface scatter data. See "Surface properties scattering tab".

80	Sets the scatter code: 0 for none, 1 for Lambertian, 2 for Gaussian, 3 for ABg, 4 for DLL, 5 for BSDF, 6 for ABg File, and 7 for IS Scatter Catalog.
81	Sets the scatter fraction, should be between 0.0 and 1.0.
82	Sets the Gaussian scatter sigma.
83	Sets the ABg file name.
84	Sets the name of the user defined scattering DLL. To set the parameters see Code 181.
85	Sets the name of the data file used by the user defined scattering DLL.
86	Sets the BSDF file name. The value should be the name of the BSDF file with no path (i.e. BrownVinyl.bsdf).
87	Sets the ABg File data file name. The value should be the name of the ABGF file with no path (e.g. SampleABGF.abgf).
88	Sets the IS Scatter Catalog data file name. The value should be the name of the ISX file with no path (e.g. BrownVinyl.ISX).
110	Sets the side for IS Scatter Catalog scattering. Use 0 for front, 1 for back.
111	Sets the sampling for IS Scatter Catalog scattering. Use 0 for 5 degrees, 1 for 2 degrees, and 2 for 1 degree.

Surface draw data. See "Surface properties draw tab".	
90	Sets the "Hide Rays To This Surface" checkbox status: 0 for off, 1 for on.
91	Sets the "Skip Rays To This Surface" checkbox status: 0 for off, 1 for on.
92	Sets the "Do Not Draw This Surface" checkbox status: 0 for off, 1 for on.
93	Sets the "Do Not Draw Edges From This Surface" checkbox status: 0 for off, 1 for on.
96	Sets the "Draw Edges As" status: 0 for squared, 1 for tapered, 2 for flat.
97	Sets "Mirror Substrate" status: 0 for none, 1 for flat, 2 for curved.
98	Sets the mirror substrate thickness value.
User defined surface scatter DLL parameters. See "Surface properties scattering tab".	
181-186	Sets the user defined scatter DLL parameters 1-6.

Changes to surface properties usually require that a subsequent UPDATE (see "UPDATE") be performed to update pupil positions, solves, and other data required for correct ray tracing. The UPDATE is not performed automatically because it is faster to execute all SETSURFACEPROPERTY keywords and then execute just one UPDATE. The function SPRO uses a very similar syntax and identical code values to "get" rather than "set" these same values.

Example:
```
! Set the glass type on surface 7 to BK7
SETSURFACEPROPERTY 7, GLAS, "BK7"
! Set the thickness of surface 2 to the thickness of surface 1
SETSURFACEPROPERTY 2, THIC, THIC(1)
! Set the value of parameter 4 on surface 11 to 7.3
SURP 11, PARM, 7.3, 4
```

Related Functions:
SPRO
Related Keywords:
SETSYSTEMPROPERTY, UPDATE

SETSYSTEMPROPERTY,SYSP
Sets properties of the system, such as system aperture, field, wavelength, and other data.
Syntax:
```
SETSYSTEMPROPERTY code, value1, value2
SYSP code, value1, value2
```

Discussion:
This keyword requires a numeric expression that evaluates to an integer code which specifies what property is being modified. The second and third arguments are the new values for the specified property, and they may be either text in quotes, a string variable, or a numeric expression, depending upon the code. For most codes, the property value being modified is defined by the value1 argument. A few operands require both a value1 and a value2, as described in the table below.
If the property being modified is under control of the Multi-Configuration Editor, then the multi-configuration data for the current configuration only will also be modified to reflect the changed property.
SYSP is a shorthand for SETSYSTEMPROPERTY and is functionally identical.

Code	Property
4	Adjust Index Data To Environment. Use 0 for off, 1 for on. See "Adjust Index Data To Environment".
10	Aperture Type code. See "Aperture Type" for code values and discussion.
11	Aperture Value. See "Aperture Value".
12	Apodization Type code. Use 0 for uniform, 1 for Gaussian, 2 for cosine cubed. See "Apodization Type".
13	Apodization Factor. See "Apodization Factor".
14	Telecentric Object Space. Use 0 for off, 1 for on. See "Telecentric Object Space".
15	Iterate Solves When Updating. Use 0 for off, 1 for on. See "Iterate Solves When Updating".
16	Lens Title. See "Lens Title".
17	Lens Notes. See "Notes".
18	Afocal Image Space. Use 0 for off, 1 for on. See "Afocal Image Space".
21	Global coordinate reference surface. See "Global Coordinate Reference Surface".
23	Glass catalog list. Use a string or string variable with the glass catalog name, such as "SCHOTT". To specify multiple catalogs use a single string or string variable containing names separated by spaces, such as "SCHOTT HOYA OHARA".
24	System Temperature in degrees Celsius. See "Temperature in degrees C".
25	System Pressure in atmospheres. See "Pressure in ATM".
26	Reference OPD method. Use 0 for absolute, 1 for infinity, 2 for exit pupil, and 3 for absolute 2. See "Reference OPD".
30	Lens Units code. Use 0 for mm, 1 for cm, 2 for inches, or 3 for Meters. Changing lens units does not scale or convert the lens data in any way, it only changes how the lens prescription data is interpreted. See "Lens Units".
31	Source Units Prefix. Use 0 for Femto, 1 for Pico, 2 for Nano, 3 for Micro, 4 for Milli, 5 for None,

	6 for Kilo, 7 for Mega, 8 for Giga, and 9 for Tera. See "Source Units".
32	Source Units. Use 0 for Watts, 1 for Lumens, and 2 for Joules. See "Source Units".
33	Analysis Units Prefix. Use 0 for Femto, 1 for Pico, 2 for Nano, 3 for Micro, 4 for Milli, 5 for None,
	6 for Kilo, 7 for Mega, 8 for Giga, and 9 for Tera. See "Analysis Units".
34	Analysis Units "per" Area. Use 0 for square mm, 1 for square cm, 2 for square inches, 3 for square Meters, and 4 for square feet.
35	MTF Units code. Use 0 for cycles per millimeter, or 1 for cycles per milliradian. See "MTF Units".
40	Coating File name. See "Coating File".
41	Scatter Profile name. See "Scatter Profile".
42	ABg Data File name. See "ABg Data File".
43	GRADIUM Profile name. See "GRADIUM Profile".
50	NSC Maximum Intersections Per Ray. See "Maximum Intersections Per Ray" .
51	NSC Maximum Segments Per Ray. See "Maximum Segments Per Ray".
52	NSC Maximum Nested/Touching Objects. See "Maximum Nested/Touching Objects".
53	NSC Minimum Relative Ray Intensity. See "Minimum Relative Ray Intensity".
54	NSC Minimum Absolute Ray Intensity. See "Minimum Absolute Ray Intensity".
55	NSC Glue Distance In Lens Units. See "Glue Distance In Lens Units".
56	NSC Missed Ray Draw Distance In Lens Units. See "Missed Ray Draw Distance in Lens Units".
57	NSC Retrace Source Rays Upon File Open. Use 0 for no, 1 for yes. See "Retrace Source Rays Upon File Open".
58	NSC Maximum Source File Rays In Memory. See "Maximum Source File Rays In Memory".
59	Simple Ray Splitting. Use 0 for no, 1 for yes. See "Simple Ray Splitting".

60	Polarization Jx. See "Jx, Jy, X-Phase, Y-Phase".
61	Polarization Jy. See "Jx, Jy, X-Phase, Y-Phase".
62	Polarization X-Phase. See "Jx, Jy, X-Phase, Y-Phase".
63	Polarization Y-Phase. See "Jx, Jy, X-Phase, Y-Phase".
64	Convert thin film phase to ray equivalent. Use 0 for no, 1 for yes. See "Convert thin film phase to ray equivalent".
65	Unpolarized. Use 0 for no, 1 for yes. See "Unpolarized".
66	Method. Use 0 for X-axis, 1 for Y-axis, and 2 for Z-axis. See "Method".
70	Ray Aiming. Use 0 for off, 1 for on, 2 for aberrated. See "Ray Aiming".
71, 72, 73	Ray aiming pupil shift x, y, and z. See "Pupil Shift, Pupil Compress".
74	Use Ray Aiming Cache. Use 0 for no, 1 for yes. See "Use Ray Aiming Cache".
75	Robust Ray Aiming. Use 0 for no, 1 for yes. See "Robust Ray Aiming (slow)".
76	Scale Pupil Shift Factors By Field. Use 0 for no, 1 for yes. See "Pupil Shift, Pupil Compress".
77, 78	Ray aiming pupil compress x, y. See "Pupil Shift, Pupil Compress".
100	Field type code. See "Fields".
101	Number of fields.
102, 103	The field number is value1, value2 is the field x, y coordinate
104	The field number is value1, value2 is the field weight
105, 106	The field number is value1, value2 is the field vignetting decenter x, decenter y
107, 108	The field number is value1, value2 is the field vignetting compression x, compression y
109	The field number is value1, value2 is the field vignetting angle
110	The field normalization method, value 1 is 0 for radial and 1 for rectangular
200	Primary wavelength number. See "Wavelengths".
201	Number of wavelengths

202	The wavelength number is value1, value 2 is the wavelength in micrometers.
203	The wavelength number is value1, value 2 is the wavelength weight
901	The number of CPU's to use in multi-threaded computations, such as optimization. If the passed value is zero, the number of CPU's will be set to the default value. When testing this value using the function SYPR, this returns the total number of CPU's available as reported by the operating system.

Usually, changes to system properties do not become effective until after the UPDATE keyword is executed.
Example:
```
! Set the number of wavelengths to 3
SETSYSTEMPROPERTY 201, 3
! Set the number of fields to 4
SYSP 101, 4
```

Related Functions:
SYPR
Related Keywords:
SETSURFACEPROPERTY, UPDATE

SETTEXTSIZE
Changes the size of the characters drawn by the GTEXT command.
Syntax:
```
SETTEXTSIZE xsize, ysize
```

Discussion:
The arguments refer to the fraction of the graphic screen width that each character represents. For example, the default text size is 70 40. This means each character is 1/70 of the graphic screen width, and 1/40 of the screen height. An argument of zero restores the text size to the default.
Example:
```
! Make text twice default size
SETTEXTSIZE 35, 20
! Restore text size to default
SETTEXTSIZE
```

SETTITLE
This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

SETTOL
Sets the data in the various columns for any tolerance operand in the tolerance data editor.
Syntax:
```
SETTOL row, column, data
```

Discussion:
The value row must evaluate to an integer expression greater than 0 and less than or equal to the current number of tolerance operands. To set the operand type. use a column value of 0. The data is either a literal string or a string variable that contains the name of the operand, such as "TTHI". To set the integer 1, 2, or 3 values, use a column number of 1, 2, or 3. To set the min or max tolerance values, use column 4 or 5, respectively. To set the "Do Not Adjust During Inverse Tolerancing" checkbox, use column 6. The data is 1 for checked, 0 for unchecked. To set the comment, use column 99. The data is either a literal string or a string variable that contains the comment. The nominal value cannot be set using SETTOL.
See also INSERTTOL and DELETETOL.
Example:

```
SETTOL 16, 0, "TRAD"
```

Related Functions:
TOLV, $TOLOPERAND, $TOLCOMMENT

SETUNITS

This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

SETVAR

Changes the state of variables for optimization.
Syntax:
```
SETVAR surf, code, status, object
or
SETVAR config, M, status, operand
```

Discussion:
Surf must evaluate to an integer between 0 and the maximum number of surfaces. Config must evaluate to an integer between 1 and the number of configurations. Code must be one of the following strings:
```
R for radius of curvature
T for thickness
C for conic
G for glass
I for glass index
J for glass Abbe
K for glass dpgf
Pn for parameter n
D for thermal coefficient of expansion
En for extra data value n
M for multi-configuration data, see discussion below
Nn for non-sequential component position data, 1-6 for x, y, z, tx, ty, tz
On for non-sequential component parameter data, where n is the parameter number
```

If status evaluates to 0, then the variable status is removed. Otherwise, the value is made variable. If the code is Nn or On; the object number must be provided; otherwise, it should be omitted. If the code is M, then the syntax for this command is as shown under "syntax" above.
Examples:
```
SETVAR j+3, R, 1
SETVAR 5, P6, 0
SETVAR surfk+2, E06, status
SETVAR config, M, status, operand
SETVAR 1, O32, 1, 5
```

SETVECSIZE

Changes the maximum size of the VEC1, VEC2, VEC3, and VEC4 arrays.
Syntax:
```
SETVECSIZE n
```

Discussion:
The expression must evaluate to an integer argument between 1 and 18,000,000. All four vector variables are always the same size. All data in the vectors will be lost during the resize. The initial size of the vectors is 1000.

SETVIG

Sets the vignetting factors for the lens.
Syntax:
```
SETVIG
```

Discussion:

See "Vignetting factors" for a description of vignetting factors.

SHOWBITMAP

Displays a BMP, JPG or PNG file in a viewer window.

Syntax:
```
SHOWBITMAP filename
```

Discussion:

This keyword requires the name of the BMP, JPG or PNG file. The extension must be included. The filename may be enclosed in quotes if any blank or other special characters are used. The file must be located in the <data>\<images> folder. This command will open a new window to display the file.

Example:
```
SHOWBITMAP "BARCHART.BMP"
```

SHOWFILE

Displays a text file to the screen using the Zemax file viewer.

Syntax:
```
SHOWFILE filename, saveflag
```

Discussion:

The filename must be a valid file name. The file must be a text file (as would be created by OUTPUT and PRINT commands in ZPL) and must be in the current folder. Once the file is displayed, it may be scrolled up and down and printed like any other text file. The ability to scroll and print the data is the primary advantage of using OUTPUT and SHOWFILE instead of PRINT commands. SHOWFILE also closes the file if no CLOSE command has been executed. If the saveflag is zero or omitted, then the file is erased when the window is closed. if saveflag is any value other than zero, then the file remains even after the window is closed.

Example:
```
OUTPUT "test.txt"
PRINT "Print this to a file."
SHOWFILE "test.txt"
```

Related Keywords:
OPEN, OUTPUT, CLOSE, PRINT, PRINTFILE

SOLVEBEFORESTOP

Enables ZPL solves to be placed on surfaces prior to the stop surface.

Syntax:
```
SOLVEBEFORESTOP
```

Discussion:

This keyword is used only for supporting ZPL Macro solves. It must be placed on the first line of the macro in order to prevent Zemax from issuing an error message. Usage of this keyword implies that the user understands when such solves are valid and when they are not.

Any solve placed prior to the stop should not be based upon ray data. For example, the RAYTRACE keyword and data that it returns should not be used. Zemax doesn't perform any checks to ensure this is the case. If the macro does contain ray-based data then the macro solve may compute erroneous values.

See "Using ZPL Macro solves".

SOLVERETURN

Returns a computed solve value back to the editor calling the solve.

Syntax:
```
SOLVERETURN x
```

Discussion:

This keyword is used only for supporting ZPL Macro solves. The macro computes a single value, and returns the value to the editor calling the solve using this keyword. If more than one SOLVERETURN keyword exists, only the last SOLVERETURN called is considered and prior calls are ignored.

If an error or invalid condition occurs in the macro, and the solve value cannot be computed, then the macro should return without calling SOLVERETURN. The absence of the SOLVERETURN call will indicate to Zemax that the solve cannot be computed and the optical system is in an error condition. This is particularly important during optimization. See "Using ZPL Macro solves".

SOLVETYPE

Changes the solve status on a given surface and value. Only some solve types are supported; contact Zemax technical support for information on setting other types of solves.

Syntax:

```
SOLVETYPE surf, CODE, arg1, arg2, arg3, arg4
```

Discussion:

Surf must evaluate to an integer argument between 0 and the maximum number of surfaces. The CODE must be a mnemonic as listed in the following table. The arg1 - arg4 expressions evaluate to the first - fourth solve parameters as specified in "SOLVES". Note that for cross-column Pickup solves, the column numbers are defined in "Integer codes for column numbers". For non-sequential pickup solves, the arguments are the first through fourth lines that follow the "Solve Type" selection on the NSC solve dialog box. Some codes do not use any or all of the arguments and the arguments may be omitted in these cases.

CODES FOR SOLVETYPE KEYWORD

Solve Type	CODE
Curvature: Fixed (turn solve off)	CF
Curvature: Variable	CV
Curvature: Marginal Ray	CM
Curvature: Chief Ray	CC
Curvature: Pickup	CP
Curvature: Marginal Ray Normal	CN
Curvature: Chief Ray Normal	CO
Curvature: Aplanatic	CA
Curvature: Element Power	CE
Curvature: Concentric With Surface	CQ
Curvature: Concentric With Radius	CR
Curvature: F/#	CG
Curvature: ZPL Macro	CZ
Thickness: Fixed (turn solve off)	TF
Thickness: Variable	TV

Thickness: Marginal Ray Height	TM
Thickness: Chief Ray Height	TC
Thickness: Edge Thickness	TE
Thickness: Pickup	TP
Thickness: Optical Path Difference	TO
Thickness: Position	TL
Thickness: Compensator	TX
Thickness: Center Of Curvature	TY
Thickness: Pupil Position	TU
Thickness: ZPL Macro	TZ
Glass: Fixed (turn solve off)	GF
Glass: Model	GM
Glass: Pickup	GP
Glass: Substitute	GS
Glass: Offset	GO
Semi-Diameter: Automatic	SA
Semi-Diameter: User Defined	SU
Semi-Diameter: Pickup	SP
Semi-Diameter: Maximum	SM
Semi-Diameter: ZPL Macro	SZ
Conic: Fixed (turn solve off)	KF
Conic: Pickup	KP
Conic: ZPL Macro	KZ
Parameter: Fixed (turn solve off). Replace "p" with the parameter number in the code, for example, PF_3 will turn off the solve on parameter 3.	PF_p
Parameter: Pickup. Replace "p" with the parameter number in the code, for example,	PP_p

PP_4 will set the solve on parameter 4.	
Parameter: Chief Ray. Replace "p" with the parameter number in the code, for example, PC_1 will set the solve on parameter 1.	PC_p
Parameter: ZPL Macro. Replace "p" with the parameter number in the code, for example, PZ_1 will set the solve on parameter 1.	PZ_p
Thermal Coefficient of Expansion: Fixed (turn solve off)	HF
Thermal Coefficient of Expansion: Pickup	HP
Extra Data Value: Fixed (turn solve off). Replace "e" with the extra data number in the code, for example, EF_3 will turn off the solve on extra data value 3.	EF_e
Extra Data Value: Pickup. Replace "e" with the extra data number in the code, for example, EP_4 will set the solve on extra data value 4.	EP_e
Extra Data Value: ZPL Macro. Replace "e" with the extra data number in the code, for example, EZ_4 will set the solve on extra data value 4.	EZ_e
Non-Sequential Component Pickup X, Y, Z, Tilt-X, Tilt-Y, Tilt-Z, Material. Replace "o" with the object number in the code, for example, NSC_PX_14 will set a pickup solve on object 14. NSC_PMAT_o is redundant with NSC_MATP_o.	NSC_PX_o, NSC_PY_o, NSC_PZ_o, NSC_PTX_o, NSC_PTY_o, NSC_PTZ_o, NSC_PMAT_o (see left)
Non-Sequential Component Material is fixed, model glass, pick up, or offset. Replace "o" with the object number in the code, for example, NSC_MATM_11 will set the material on object 11 to be a model glass. NSC_MATP_o is redundant with NSC_PMAT_o.	NSC_MATF_o, NSC_MATM_o, NSC_MATP_o, NSC_MOFF_o (see left)
Non-Sequential Component ZPL Macro solve on X, Y, Z, Tilt-X, Tilt-Y, Tilt-Z. Replace "o" with the object number in the code, for example, NSC_ZX_14 will set a macro solve on object 14.	NSC_ZX_o, NSC_ZY_o, NSC_ZZ_o, NSC_ZTX_o, NSC_ZTY_o, NSC_ZTZ_o (see left)
Non-Sequential Component Parameter Pickup. Replace "o" with the object number and "p" with the parameter number in the code, for example, NSC_PP_11_7 will set a pickup solve on object 11, parameter 7.	NSC_PP_o_p (see left)
Non-Sequential Component ZPL Macro solve. Replace "o" with the object number and "p" with the parameter number in the code, for example,	NSC_ZP_o_p (see left)

NSC_ZP_11_7 will set a macro solve on object 11, parameter 7.	

Example:
```
! The following line will add a glass pickup solve
! on surface 7, picking up from surface 5:
SOLVETYPE 7, GP, 5
! Add a thickness pickup with a scale factor of -1:
SOLVETYPE 7, TP, 5, -1

! Set a pickup solve on surface 1, NSC object 12 Z position,
! pick up from object 11, with a scale factor of 2, offset 3,
! from the parameter 7 column. Note the column number is argument 4.
! The column number is 0 for the same column, 1-6 for x, y, z, tilt x, tilt y, tilt z,
! respectively. The column number for the parameter columns
! is 6 + the desired parameter number.
! In summary, the syntax is:
! SOLVETYPE, surf, code, object, scale, offset, column
! where code has the object/parameter number embedded as shown in the table above.
! The syntax for this example is:
SOLVETYPE 1, NSC_PZ_12, 11, 2, 3, 13
```

Related Functions:
SOLV

STOPSURF
STOPSURF sets the current stop surface location by number.
Syntax:
```
STOPSURF surf
```

Discussion:
The UPDATE command must be issued before the new data takes effect.
Example:
```
STOPSURF n+2
```

Related Keywords:
UPDATE

SUB
See GOSUB.

SURFTYPE
This command is obsolete. See "SETSURFACEPROPERTY, SURP".

SUSPENDUPDATES

See also RESUMEUPDATES. Prevents any UI editor windows from being updated by ZPL commands while suspended. This command can greatly speed up macro execution when performing many updates to the system, such as inserting hundreds or thousands of merit function rows. The number of suspend and resume calls are reference counted such that no updates will be performed until RESUMEUPDATES is encountered as many times as SUSPENDUPDATES (or the macro completes execution). Note that changes will still be made to the internal lens state, however the user interface will not reflect the changes. Also note that the UI is not automatically updated when RESUMEUDPATES is called; to do so, use the UPDATE EDITORS command.

TELECENTRIC
This command is obsolete. See "SETSYSTEMPROPERTY, SYSP".

TESTPLATEFIT

TESTPLATEFIT calls the test plate fitting routine. See "Test Plate Fitting".

Syntax:
```
TESTPLATEFIT tpd_file, log_file, method, number_cycles
```

Discussion:

This keyword requires string expressions for the test plate data file, and the name of a file for the output log file. The method is an integer between 0 and 4, inclusive, for try all methods, best to worst, worst to best, long to short, and short to long, respectively. The integer number_cycles is 0 for automatic or the maximum number of optimization cycles to execute. Note the tpd_file name should NOT include the path, since all test plate files are in a fixed folder; while the path should be included for the log file.

This keyword may take a long time to execute. It is advisable to display the log file after completion of the fitting, or use some other means to indicate when the fitting is complete.

Example:
```
TESTPLATEFIT "optico.tpd", "c:\temp\logfile.dat", 0, 0
SHOWFILE "c:\temp\logfile.dat"
```

THIC

This command is obsolete. See "SETSURFACEPROPERTY, SURP".

TIMER

Resets the internal clock. This feature is used in conjunction with the ZPL function ETIM() for measuring the elapsed time since the last TIMER command.

Syntax:
```
TIMER
```

Discussion:

TIMER and ETIM() are used primarily for testing execution efficiency of the ZPL interpreter and various program architectures.

Example:
```
i = 0
TIMER
LABEL 1
x = RAND(1000)
i = i+1
if i < 10000 THEN GOTO 1
FORMAT .1
PRINT "Elapsed time:", ETIM(), "Seconds"
```

TOLERANCE

Runs a tolerance analysis and saves the tolerance report to a text file.

Syntax:
```
TOLERANCE top_file_name, out_file_name
```

Discussion:

TOLERANCE runs a tolerance analysis on the current lens based upon the settings in the tolerance options file defined by top_file_name. This file name should end in the extension TOP and be placed in the <data>\Configs folder (see "Folders"). TOP files are created by saving the tolerance options from the tolerance dialog, see "Other buttons". The out_file_name will be a text file (any extension may be used) and the file will be placed in the same folder as the current lens file. Neither top_file_name nor out_file_name should specify a path.

UNLOCKWINDOW

Unlocks any one or all locked windows.

Syntax:
```
UNLOCKWINDOW winnum
```

Discussion:
See "Graphic windows operations". If the winnum is zero, then all open windows are unlocked. If the winnum argument is -1, then the currently executing window will be unlocked at the end of the macro execution.
Example:
```
UNLOCKWINDOW 2
```

Related Keywords:
LOCKWINDOW

UPDATE

Updates pupil positions, index data, paraxial constants, semi-diameters, maximum field normalization values, solves, non-sequential objects, and other data required for both sequential and non-sequential ray tracing. The UPDATE keyword MUST be used before tracing or evaluating a system if the prescription data, such as surface type, radii, thicknesses, system apertures, wavelengths, or non-sequential object types or parameters, have been altered since the last UPDATE.
If the UPDATE command is followed by the keyword "ALL", then all open windows, except ZPL macro windows, will be updated as well. If the UPDATE command is followed by an expression which evaluates to an integer corresponding to an open window, then the specified window is updated, as long as it is not the window of the macro currently executing. If UPDATE is followed by EDITORS, any open spreadsheet editors will be refreshed to show recent changes in the lens data. If UPDATE is followed by the keyword MACROS, then all open ZPL macro windows other than the calling window are updated.
Syntax:
```
UPDATE
UPDATE ALL
UPDATE n
UPDATE EDITORS
UPDATE MACROS
```

VEC1, VEC2, VEC3, VEC4

These keywords are used to set the vector variables VEC1, VEC2, VEC3, and VEC4. Each vector can store an array of double-precision floating point numbers.
Syntax:
```
VEC1 subscript, value
VEC2 subscript, value
VEC3 subscript, value
VEC4 subscript, value
```

Discussion:
VEC1..4 are used to store data in an array. The subscript value can be any expression which is rounded down to an integer. The resulting integer expression must be between 0 and the current maximum vector size, which is initially 1000. The ZPL functions VEC1..4 can be used to extract the data. To change the array size use the SETVECSIZE keyword.
Example:
```
i = 0
LABEL 1
i = i + 1
VEC1 i, i
IF i < 10 THEN GOTO 1
j = 0
LABEL 2
j = j + 1
VEC2 j, VEC1(j) * VEC1(j)
IF j < 10 THEN GOTO 2
i = 0
LABEL 3
i = i + 1
PRINT "x = ", VEC1(i), " x*x = ", VEC2(i)
IF i < 10 THEN GOTO 3
```

```
PRINT
PRINT "All done!"
```

WAVL,WWGT

These commands are obsolete. See "SETSYSTEMPROPERTY, SYSP".

XDIFFIA

Computes the Extended Diffraction Image Analysis feature and saves the result to a ZBF file. For a description of the ZBF (Zemax Beam File) format see "Zemax Beam File (ZBF) binary format". For a description of the Extended Diffraction Image Analysis feature see "Extended Diffraction Image Analysis".
Syntax:
```
XDIFFIA outfilename, infilename
```

Discussion:
This keyword requires the name of the output ZBF file, and optionally, the name of the input IMA or BIM file. If the extension to the outfilename is not provided, the extension ZBF will be appended. The extension must be provided on the infilename. The filenames must be enclosed in quotes if any blank or other special characters are used. The outfilename will be placed in the <pop> folder. The infilename must be placed in the <data>\<im- ages> folder. No paths should be provided with the file names.
The settings for the Extended Diffraction Image Analysis feature will be those settings previously saved for the current lens. To make adjustments to the settings, open an Extended Diffraction Image Analysis window, choose the appropriate settings, then press "Save". All subsequent calls to XDIFFIA will use the saved settings. The exceptions are the output file name, which is specified as the first argument after the XDIFFIA keyword, and the input source file, which is optionally specified as the second argument after the XDIFFIA keyword.
Example:
```
XDIFFIA "output will be in this file.ZBF", "SOMEIMAFILE.IMA"
```

Related Functions:
ZBFCLR, ZBFSHOW, ZBFSUM, ZBFMULT

ZBF2MAT

Converts a ZBF file to a Matlab MAT file.
Syntax:
```
ZBF2MAT zbffilename, matfilename, surfacerange, integer1, integer2
```

Discussion:
The first two arguments are strings giving the full name including path and extension of the ZBF file to be converted and the MAT file to be created, respectively. The surfacerange argument is an integer flag indicating whether a ZBF file at an individual surface is to be converted or a range of ZBF files, one for a given surface, are to be converted. If this flag is set to zero then integer1 and integer2 can be ignored. If surfacerange is non-zero integer1 is the starting surface to be converted and integer2 is the final surface to be converted. Note that there are restrictions on the ZBF file name as described in "MAT File Generator".

ZBFCLR

Clears the complex amplitude data in a ZBF file.
Syntax:
```
ZBFCLR filename
```

Discussion:
This keyword requires only the name of the ZBF file.
General comments about all ZBF related keywords
For a description of the ZBF (Zemax Beam File) format see "Zemax Beam File (ZBF) binary format". Zemax requires the filename extension to be ZBF, and will append or replace the extension as required. The filename may be enclosed in quotes if any blank or other special characters are used. The file must be located in the <pop> folder. All ZBF output files will be placed in this same folder.

Example:
```
ZBFCLR "some beam file name.ZBF"
ZBFCLR N$
```


ZBFMULT

Multiplies the complex amplitude data in a ZBF file by a complex factor.

Syntax:
```
ZBFMULT filename, Ax, Bx, Ay, By
```

Discussion:

This keyword requires the name of the ZBF file and the real (A) and imaginary (B) parts of the complex number to multiply every point in the ZBF file by. There are separate A and B for the x and y polarized light directions. The resulting data is written back to the same file name.

See "Purpose" for comments that apply to all ZBF related keywords.

Example:
```
ZBFMULT "some beam file name.ZBF", 0.0, 1.0, 0.0, -1.0
```


ZBFPROPERTIES

Opens the specified ZBF file and places various data about the beam in a vector variable.

Syntax:
```
ZBFPROPERTIES filename, vector
```

Discussion:

This keyword requires the name of the ZBF file and a value for the vector number to place the data in. The value of vector must be between 1 and 4, inclusive. After the ZBFPROPERTIES function executes, the following beam data will be placed in the specified vector: nx, ny, dx, dy, waist_x, waist_y, position_x, position_y, rayleigh_x, rayleigh_y, wavelength (in lens units), total power, peak irradiance (power per area), the is_polarized flag (0 for no, 1 for yes), and the media index; the values are placed in vector positions 1 through 15.

See "Purpose:" for comments that apply to all ZBF related keywords.

Example:
```
ZBFPROPERTIES "TEST.ZBF", 1
rayleighx = VEC1(9)
```


ZBFREAD

Opens the specified ZBF file and places the electric field and beam property data in two user-defined array variables.

Syntax:
```
ZBFREAD filename, beamname, propertyname
```

Discussion:

This keyword requires the name of the ZBF file and the name of two arrays defined by a previous call to DECLARE. The beamname must be a 3 dimensional array, of minimum size (nx, ny, 2) for an unpolarized beam and minimum size (nx, ny, 4) for a polarized beam. The propertyname array must be a one dimensional array of minimum size 14. After the ZBFREAD function executes, the following beam data will be placed in the specified propertyname array: nx, ny, dx, dy, waist_x, waist_y, position_x, position_y, rayleigh_x, rayleigh_y, wavelength (in lens units), total power, peak irradiance (power per area), the is_polarized flag (0 for no, 1 for yes), and the media index; the values are placed in array positions 1 through 15. The electric field data will be placed in the beamname array. The third dimension of the beamname array is 1 for Ex Real, 2 for Ex Imaginary, and if the beam is polarized, 3 for Ey Real, and 4 for Ey Imaginary.

See "Purpose:" for comments that apply to all ZBF related keywords. See also "ZBFWRITE".

Example:
```
! First get the beam size
ZBFPROPERTIES "TEST1.ZBF", 1
nx = vec1(1)
ny = vec1(2)
```

```
ip = vec1(14) ! The "is polarized" flag

! Allocate enough memory to hold the beam
IF (ip == 0) THEN DECLARE B, DOUBLE, 3, nx, ny, 2
IF (ip == 1) THEN DECLARE B, DOUBLE, 3, nx, ny, 4
DECLARE P, DOUBLE, 1, 20
ZBFREAD "test1.zbf", B, P

FOR j, 1, ny, 1
     FOR i, 1, nx, 1
             FORMAT 4.0
             PRINT i, j,
             FORMAT 12.6
             IF (ip == 1)
             PRINT B(i, j, 1),
             PRINT B(i, j, 2),
             PRINT B(i, j, 3),
             PRINT B(i, j, 4)
             ELSE
             PRINT B(i, j, 1),
             PRINT B(i, j, 2)
             ENDIF
     NEXT
NEXT

! save the beam
ZBFWRITE "TEST2.ZBF", B, P

! release the allocated memory
RELEASE B
RELEASE P
```

ZBFRESAMPLE

Resamples a ZBF file to a new width and point spacing.
Syntax:
```
ZBFRESAMPLE filename, nx, ny, wx, wy, decenterx, decentery
```

Discussion:
This keyword requires the name of the ZBF file and six numbers. The beam will be resampled and interpolated as required to create a new beam file with nx and ny points, of total width wx and wy, in the x and y directions respectively. The nx and ny values must be powers of 2; such as 32, 64, 128, etc. The decenterx and decentery values may be provided to optionally decenter the new beam relative to the old beam. If either nx or ny is zero, no change is made to the existing beam sampling. If either wx or wy is zero, no change is made to the existing beam width. The length units in the ZBF file are converted automatically to the current lens units. The resulting data is written back to the same file name.
Example:
```
ZBFRESAMPLE "TEST.ZBF", 128, 128, 25.4, 25.4, 0, 0.4
```

ZBFSHOW

Displays a ZBF file in a viewer window.
Syntax:
```
ZBFSHOW filename
```

Discussion:
This keyword requires only the name of the ZBF file. This command will open a new window to display the beam file.
Example:
```
ZBFSHOW "new beam data.ZBF"
ZBFSHOW N$
```

ZBFSUM

Sums either coherently or incoherently the data in two ZBF files and places the resulting data in a third ZBF file.

Syntax:
```
ZBFSUM coherent, filename1, filename2, outfilename
```

Discussion:

This keyword requires an integer to indicate if the sum is coherent (any value other than zero) or incoherent (zero), and the names of three ZBF files. If an incoherent sum is performed, then the output data will be real valued only. If the two source files do not have the same number of data points, point spacing, and reference radii in both x and y directions, then the second source file listed is first scaled and interpolated, and the reference radii is adjusted, to match the first file before the sum is performed. The length units in the ZBF files are converted automatically to the current lens units. The outfilename may be the same as one of the source file names; in which case the original file is overwritten.

Example:
```
ZBFSUM 1, "a.ZBF", "b.zbf", "coherent a plus b.zbf"
ZBFSUM 0, "a.ZBF", "b.zbf", "incoherent a plus b.zbf"
ZBFSUM 0, A$, B$, C$
```

ZBFTILT

Multiplies the data in a ZBF file by a complex phase factor to introduce phase tilt to the beam.

Syntax:
```
ZBFTILT filename, cx, cy, tx, ty
```

Discussion:

This keyword requires the name of the ZBF file and four numbers. The phase of the beam is modified by a phase angle given by $\theta = (x - cx)tx + (y - cy)ty$. The cx and cy values are the center of the phase tilt, and tx and ty are the slopes of the tilt in units of radians per lens unit length. The coordinates x and y refer to positions within the beam file, with the center coordinate (x = 0, y = 0) being at the point (nx/2 + 1, ny/2 + 1) where nx and ny are the number of points in the x and y directions. The length units in the ZBF file are converted automatically to the current lens units. The resulting data is written back to the same file name.

Example:
```
ZBFTILT "TEST.ZBF", 0.0, 0.0, 0.0, 0.01237
```

ZBFWRITE

Writes electric field and beam property data arrays to a ZBF file.

Syntax:
```
ZBFWRITE filename, beamname, propertyname
```

Discussion:

This keyword requires the name of the ZBF file and the name of two arrays defined by a previous call to DECLARE. The beamname must be a 3 dimensional array, of minimum size (nx, ny, 2) for an unpolarized beam and minimum size (nx, ny, 4) for a polarized beam. The propertyname array must be a one dimensional array of minimum size 14. The following beam data must be placed in the specified propertyname array: nx, ny, dx, dy, waist_x, waist_y, position_x, position_y, rayleigh_x, rayleigh_y, wavelength (in lens units), total power, peak irradiance (power per area), the is_polarized flag (0 for no, 1 for yes), and the media index; the values are placed in array positions 1 through 15. The electric field data must be placed in the beamname array. The third dimension of the beamname array is 1 for Ex Real, 2 for Ex Imaginary, and if the beam is polarized, 3 for Ey Real, and 4 for Ey Imaginary.
See also "ZBFREAD".

ZRD2MAT

Converts a ZRD file to a Matlab MAT file.

Syntax:
```
ZRD2MAT zrdfilename, matfilename, savesegment, integer1, integer2
```

Discussion:

The first two arguments are strings giving the full name including path and extension of the ZRD file to be converted and the MAT file to be created, respectively. The next argument, which is savesegment, is an integer flag indicating whether an individual segment in the ZRD file is to be converted or a range of rays. If this flag is set to zero then integer1 is the number of the first ray in the range to be converted and integer2 is the number of the last ray. If savesegment is non-zero then only one segment is converted; in this case integer1 is the ray number and integer2 is the segment number.

ZRDAPPEND

Appends data from one ZRD file onto the end of a second file. The first file is left unaltered. This command only makes sense to use if the two ZRD files were traced in the same system, as no attempt is made to modify object numbers or coordinates.

Syntax:

```
ZRDAPPEND infilename, outfilename
```

Discussion:
This keyword requires the name of two input ZRD files. For information on the ZRD format, see "Ray database (ZRD) files". The file names may each be defined by either a literal string or a single string variable. The file names must include the full path name. Although any extensions may be specified, the extension ZRD is recommended for both file names. The input and output file names must be distinct. The input and output files must use the same ZRD format. See also ZRDSUM.

Example:

```
ZRDAPPEND "C:\TEMP\CART.ZRD", "C:\TEMP\HORSE.ZRD"
```

ZRDFILTER

Opens a ZRD ray database file, applies a filter, and saves the filtered subset of rays to a new ZRD file.

Syntax:

```
ZRDFILTER infilename, outfilename, filterstring
```

Discussion:
This keyword requires the name of an input ZRD file, the name of the output (filtered) ZRD file, and the filter string to apply. For information on the ZRD format, see "Ray database (ZRD) files". For information on the filter string format see "The filter string". The file names and filter string may each be defined by either a literal string or a single string variable. The file names must include the full path name. Although any extensions may be specified, the extension ZRD is recommended for both the input and output files. The input and output file names must be distinct.

Example:

```
ZRDFILTER "C:\TEMP\TEST1.ZRD", "C:\TEMP\TEST1_H4.ZRD", "H4"
```

ZRDPLAYBACK

This keyword reads a ZRD file, and adds ray amplitude data to one or all detectors. This command only makes sense to use if the ZRD file was generated with the currently loaded system, as no attempt is made to modify object numbers or coordinates.

Syntax:

```
ZRDPLAYBACK zrdfilename, surface, detector, clear, filterstring
```

Discussion:
This keyword requires the name of a ZRD file. For information on the ZRD format, see "Ray database (ZRD) files". The file name may each be defined by either a literal string or a single string variable. The file name must not include a path and the file must be in the same folder as the current lens file. The surface argument is the surface number of the Non-Sequential surface, use 1 for Non-Sequential mode. The integer detector is either 0 for all detectors, or the object number of a specific single detector. If clear is set to anything other than zero, then the specified detector(s) are cleared prior to playback. The optional filterstring is applied to the rays before being played back.

Example:

```
ZRDPLAYBACK "MYRAYS.ZRD", 1, 0, 1, "H3 & H5"
```

ZRDSAVERAYS

Opens a ZRD ray database file, applies a filter, and saves the rays in the filtered subset that intersect a specified object number.

Syntax:
```
ZRDSAVERAYS infilename, outfilename, filterstring, object
```

Discussion:
This keyword requires the name of an input ZRD file, the name of the output source ray DAT file, the filter string to apply, and the object number. For information on the ZRD format, see "Ray database (ZRD) files. Source ray files are described in "Source File". For information on the filter string format see "The filter string" . The file names and filter string may each be defined by either a literal string or a single string variable. The file names must include the full path name. Although any extensions may be specified, the extension ZRD is recommended for the input file, and the extension DAT is recommended for the output file. The input and output file names must be distinct. If no filtering is desired, the filterstring argument should be replaced with two double quotes like this: "".

Example:
```
ZRDSAVERAYS "C:\TEMP\TEST1.ZRD", "C:\TEMP\TEST1_H6.DAT", "", 6
```

ZRDSUM

Concatenates two ZRD files into a third file. This command only makes sense to use if the two ZRD files were traced in the same system, as no attempt is made to modify object numbers or coordinates.

Syntax:
```
ZRDSUM infilename1, infilename2, outfilename
```

Discussion:
This keyword requires the name of two input ZRD files and the name of the output ZRD file. For information on the ZRD format, see "Ray database (ZRD) files". The file names may each be defined by either a literal string or a single string variable. The file names must include the full path name. Although any extensions may be specified, the extension ZRD is recommended for all three file names. The input and output file names must be distinct. Both input files must use the same ZRD format, and the output file will use this same ZRD format. See also ZRDAPPEND.

Example:
```
ZRDSUM "C:\TEMP\TEST1.ZRD", "C:\TEMP\TEST2.ZRD", "C:\TEMP\TEST_SUM.ZRD"
```

Example Macro 1

The following sample macro will print out the image surface intercept coordinates of the chief ray at every defined field angle:
```
nfield = NFLD()
maxfield = MAXF()
n = NSUR()
FOR i, 1 , nfield, 1
      hx = FLDX(i)/maxfield
      hy = FLDY(i)/maxfield
      PRINT "Field number ", i
      RAYTRACE hx,hy,0,0,PWAV()
      PRINT "X-field angle : ",FLDX(i)," Y-field angle : ", FLDY(i)
      PRINT "X-chief ray   : ",RAYX(n), " Y-chief ray   : ", RAYY(n)
      PRINT
NEXT
PRINT "All Done!"
```

The first line of the macro calls the NFLD() function which returns the number of defined field angles, and assigns it to the variable "numfield". The second line calls MAXF(), which returns the maximum radial field and stores the value in the variable "maxfield". The number of surfaces is then stored in the variable "n" by calling the function NSUR().
The FOR loop is then defined with i being the counter for the field position, starting at 1, with a maximum value of nfield, and an increment of 1 on each loop. The two macro lines which define "hx" and "hy" use the functions FLDX() and FLDY(), which return the x- and y- field values for the current field position number "i". The chief ray is then traced by

---

the keyword RAYTRACE. Note the chief ray intersects the center of the pupil, which is why the two pupil coordinates are both zero. The PWAV() function returns the primary wavelength number, which is usually used for the chief ray. The various PRINT commands will output the chief ray coordinates on the image surface to the screen.

Example Macro 2

The following sample macro will estimate the RMS spot radius (on axis) of the current optical system. The macro traces many random rays through the system, and records the radial offset from the primary wavelength chief ray. The current wavelength weighting is applied to estimate the RMS spot radius.

```
PRINT "Primary wavelength is number ",
FORMAT .0
PRINT PWAV(),
FORMAT .4
PRINT " which is ", WAVL(PWAV()), " micrometers."
PRINT "Estimating RMS spot radius for each wavelength."

! How many random rays to trace to make estimate?
n = 100

! Initialize the timer
TIMER

! Store the number of surfaces for later use
ns = NSUR()

! Start at wavelength 1
weightsum = 0
wwrms = 0
FOR w, 1, NWAV(), 1
        rms = 0
        FOR i, 1, n, 1
                hx = 0
                hy = 0
                angle = 6.283185 * RAND(1)
                ! SQRT yields uniform distribution in pupil
                radius = SQRT(RAND(1))
                px = radius * COSI(angle)
                py = radius * SINE(angle)
                RAYTRACE hx, hy, px, py, w
                x = RAYX(ns)
                y = RAYY(ns)
                rms = rms + (x*x) + (y*y)
        NEXT
        rms = SQRT(rms/n)
        wwrms = wwrms + ( WWGT (w) * rms )
        weightsum = weightsum + WWGT(w)
        FORMAT .4
        PRINT "RMS spot radius for ", WAVL(w),
        FORMAT .6
        PRINT " is ", rms
NEXT
wwrms = wwrms / weightsum
PRINT "Wavelength weighted rms is ", wwrms
FORMAT .2
t = ETIM()
PRINT "Elapsed time was ",t," seconds."
```

Note the use of the trailing commas in the first two PRINT commands. These allow the data printed from the first three PRINT commands to appear on the same line. The intervening FORMAT commands changes the way the numerical output is printed, even on the same line. The "!" character is used to indicate a comment, which is ignored when the macro is run.

Calling a Macro from within a Macro

To call another ZPL macro from within a ZPL macro, use the CALLMACRO keyword. Data may also be passed between the macros using the keywords CALLSETDBL and CALLSETSTR and the functions CALD and $CALLSTR. The first macro that is executed, usually from the ZPL Macro Dialog box, is the "parent" macro. The parent macro can call other macros, and these macros are called "child" macros. The parent macro creates a buffer of 51 numeric and 51 string values. The buffer can be used to set or retrieve numeric or string values, which allows macros to share data, or pass arguments to one another. The buffer is unique to the parent macro, and any child macros called by the parent. If more than one parent macro is executing in parallel (for example, in two windows each updating a different macro) each parent has their own buffer, and no data is shared between parents.

The easiest way to see how the parent and child macros work is by example. This example will use just two very simple macros, but there is no hard limit as to how many macros may be called or nested.

Consider two macros, called PARENT.ZPL and CHILD.ZPL The PARENT.ZPL macro is:

```
CALLSETDBL 1, 3.5
CALLSETSTR 1, "Hello World"
CALLMACRO CHILD.ZPL
PRINT CALD(1)
```

The CHILD.ZPL macro is:

```
PRINT "Executing child macro"
PRINT CALD(1)
A$ = $CALLSTR(1)
PRINT A$
CALLSETDBL 1, 7.11
```

When the PARENT.ZPL macro is executed, the macro uses CALLSETDBL to place the numeric value 3.5 in the parent's call numeric buffer in index position 1. The CALLSETSTR is then used to place the string "Hello World" in the parent's call string buffer in index position 1. Any index between 0 and 50, inclusive may be used for either the numeric or string buffers.

The PARENT.ZPL macro then executes the macro CHILD.ZPL. The child macro prints the message "Execut- ing child macro" to the child's output window, which is not visible during execution. The child then extracts the numeric value from numeric buffer at index 1 using CALD(1), and the string value from the string buffer index 1 using $CALLSTR(1). These values are also printed. Finally, the child macro overwrites the numeric value in index 1 with the new value 7.11, and the child macro completes.

Control now returns to the parent macro, which copies the child macro output window contents to the parent's output window, and the child's output window contents are discarded. Lastly, the modified value from the numeric call buffer at index 1 is printed. The output of this macro looks like this:

```
Executing child macro
3.5000
Hello World
7.1100
```

The sample macros PARENT.ZPL and CHILD.ZPL are included with Zemax in the Macros folder.

Using ZPL Macro Solves

ZPL Macro solves call a user defined ZPL Macro to determine the solve value. For example, see the keyword "SOLVEBEFORESTOP". For more information on solves see the "Solves" section of the Lens Data Editor in the Setup Tab.

Macro solves work by invoking a user defined ZPL Macro to compute the solve. Any numerical value that can be computed in a macro can be returned to the editor calling the solve macro. The macro may make use of the data that exists elsewhere in the editors, such as previous surfaces. Once the data is computed, the macro uses the keyword SOLVERETURN to pass the data back to the editor.

For a simple example, here is a macro that computes the first -order optical power of the interface between surfaces 1 and 2, and returns this power as the solve value:

```
n1 = INDX(1)
n2 = INDX(2)
c2 = CURV(2)
```

```
SOLVERETURN (n2-n1)*c2
```

There is a disadvantage to referring to specific surfaces directly by surface number as this macro does. If new surfaces are inserted or deleted, the macro will need to be edited to reflect the new surface numbers. Also, certain features, such as mirror substrate drawing, will not work correctly if macro solves refer to fixed surface numbers. The solution is to use the ZPL function SURC to find surfaces with specific text in the comment column of the Lens Data Editor. Suppose the comment "My Surface" was placed on the desired first surface in the Lens Data Editor. The macro could support the power computation above with this revised macro:

```
A$ = "My Surface"
SURF = SURC(A$)
n1 = INDX(SURF)
n2 = INDX(SURF+1)
c2 = CURV(SURF+1)
SOLVERETURN (n2-n1)*c2
```

The surface or object number from which the macro solve is being called can also be extracted using the numeric ZPL SOSO function.
Use surface comments and the SURC function in solve macros whenever possible.

Important considerations for ZPL Macro solves

Macro solves are very general, allowing virtually any computation to be used to determine a solve value. All functions and keywords supported by ZPL are available. There are no differences between ZPL macros executed from the macro menu or executed as a solve. However, many ZPL keywords and some functions should not be used in a macro solve. For example, calling "UPDATE" from the solve will update all solves, which will invoke the macro again, resulting in an infinite loop. Other keywords, such as INPUT, will require the user to enter data every time the solve is called, which may be an enormous number of times. Other keywords that set values in the editors or create merit functions should never be used. In general, macro solves should be kept short and simple, avoid lengthy computations, and should not make modifications of any lens data. Macro solves should also never depend upon data in the editors that is subsequent to the solve; as this may also create erroneous data if the solve is called first and then the source data is modified by a separate, subsequent solve. Zemax makes no attempt to qualify or validate the solve macro. This feature offers great power and flexibility, but must be used carefully.

Macro solves may be placed prior to the stop surface only if the computations in the macro aren't based upon any ray data. Care should be taken to ensure that this is indeed the case. See the description of the ZPL keyword "SOLVEBEFORESTOP" for details.

If an error or invalid condition occurs in the macro, and the solve value cannot be computed, then the macro should return without calling SOLVERETURN. The absence of the SOLVERETURN call will indicate to Zemax that the solve cannot be computed and the optical system is in an error condition. This is particularly important during optimization.

Integer codes for column numbers

To set a Pickup solve from the ZPL macro language, the column number is required. The column number is defined as follows:

0: The current column, which is the column the solve is placed in. This is the default for all pickup solves.
1-4: Radius, Thickness, Conic, and Semi-Diameter, respectively.
5-17: Parameter 0 through 12, respectively

Extensions Group

Extensions is a section of the Programming Tab.
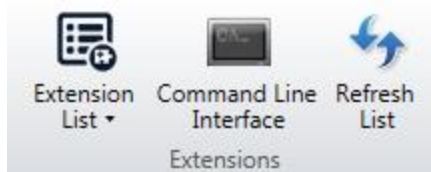
For more information on extensions, please see "About Extensions".

Extension List

The Extension List button is found in the Extensions section of the Programming Tab.
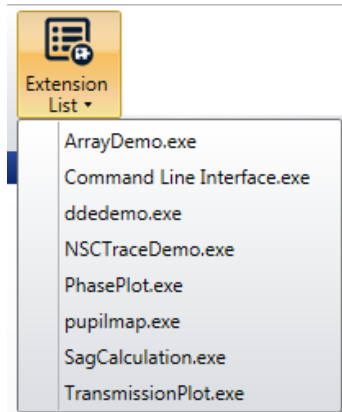
This feature shows a drop-down list of all Zemax extensions in the extensions folder is shown.

Clicking on the extension name will execute that extension.

Extensions included with Zemax

ArrayDemo Creates a table of rays traced for the current lens. This example program includes source code and illustrates the use of array ray tracing commands.

DDE_DEMO Creates a text listing of ray and system data. This example includes source code and illustrates the basic use of DDE features.

PhasePlot Creates a graph showing the phase and the inverse of the phase slope for Binary 2 surfaces. The plot indicates the phase over the semi-diameter of the surface.

Pupil Map Maps an x-y grid of rays from the entrance pupil or a ray-aimed stop surface to another surface in the system.

SagCalculation Creates an extended listing of sag data for non-radial symmetric surfaces.

TransmissionPlot Computes transmission vs. Wavelength.

Command Line Interface

The Command Line Interface is found in the Extensions section of the Programming Tab.
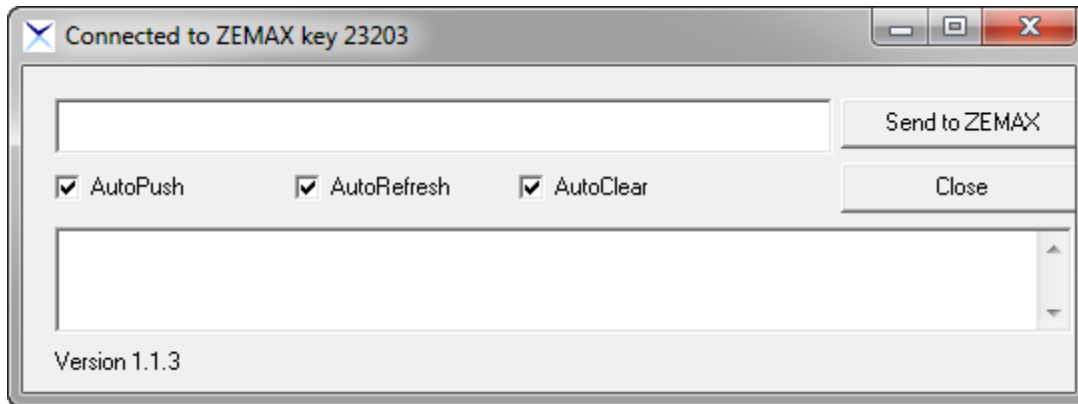
This tool opens the command line interface, and is a useful tool for testing and debugging extension commands.

Refresh List

The Refresh extensions list button is found in the Extensions section of the Programming Tab.
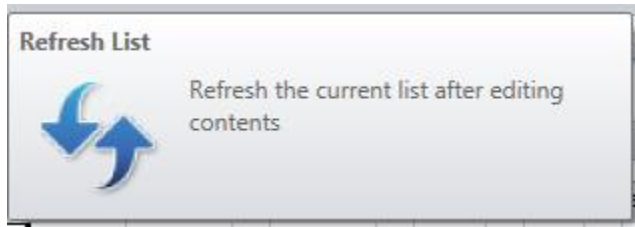
This feature updates the current extensions list; which may be required if any extensions have been added or deleted since the last time the list was refreshed. Any new extensions must be placed in the<program>\Extend folder (see "Folders").

About Extensions

This section is about using Extensions with Zemax. If you are viewing this in an HTML dialog, expand the header in Contents tab to view the subsections.

Introduction

Zemax has a very powerful feature which allows another Windows program to establish a communication link with Zemax, and for this other program to get data from Zemax on the lens. The idea is that a program can use Zemax to trace rays through the lens, and then the data is sent to the other program for further analysis or computation. Extensions are more complicated than macros; for simple programming jobs see "ZEMAX PROGRAMMING LANGUAGE".

There are three closely related ways this feature can be used to extend the capabilities of Zemax:

First, any stand-alone Windows program may be designed to establish a link with Zemax which is used to extract data about a lens that Zemax can provide- typically ray trace or other optical data. The program may use this data any way it sees fit.

Second, a stand alone program can do its analysis "hidden" from the user, and then the data generated may be displayed in the usual Zemax graphic or text window. When using this mode, the program is called an "Extension" to Zemax. Extensions appear as menu options under the "Extension List" in the Extensions section of Programming Tab of Zemax. The Extensions must be placed in the <program>\Extend folder (see "Folders") for Zemax to run them.

Third, the stand alone program may be used to compute data that Zemax can optimize. In this mode, the program is called a "User Defined Operand" or UDO. These UDO's must be placed in the <program>\UDO folder (see "Folders") for Zemax to run them.

Applications that require complex user interfaces, with many inputs, menus, and buttons are better suited to be stand alone applications that do not use Zemax to display the computed data. Applications that may be complex computationally, but require only a single dialog box type user interface, are better suited to be integrated with Zemax as Extensions. Extensions have the advantage of looking and acting just like regular Zemax analysis windows. Features implemented as Extensions may be viewed within Zemax, updated, have settings changed, and be printed, copied to the clipboard, or be panned and zoomed just like any Zemax analysis window.

UDO's are for applications that compute data that must be optimized by the Zemax optimizer. UDO's are described in detail in "Optimization" under "User defined operands".

Well written applications may even be designed to operate either as a stand alone program or as an Extension, as shown in the examples section.

The communication between application and Zemax is accomplished using Dynamic Data Exchange (DDE). DDE is a protocol defined within the Windows operating system for sharing data between programs. Two programs can establish a DDE link, with one program acting as the "server" and the other the "client". The client generally requests specific data from the server, and the server sends the data back to the client.

Zemax is intended to be the server, and any Windows program with the right modifications can be made a client.

Requirements for writing Extensions
The purpose of this chapter and the DDE features is to assist experienced programmers in writing code that can interface to Zemax. This feature allows programmers to write third party software programs that work in conjunction with Zemax. The third party program may use Zemax as a ray trace engine, but provide an interface, graphics, input and output, and extended calculations that are customized for a particular type of analysis.
Extensions can be easily implemented using any program or programming language that supports DDE. Examples of such programs include Visual Basic, Excel, and MatLab, but there are many others and DDE is a fairly common interface between scientific programs. If you are writing an application in C, then using this feature requires a knowledge of Windows and C programming, with at least some mastery over message passing, message loops, memory management, and global atoms, handles, and pointers. An excellent book that contains all the information needed to learn DDE programming is "Programming Windows, Fourth Edition" by Charles Petzold, Microsoft Press. The examples and code sections in the text of this chapter are not complete programs, but are code fragments intended only for illustration. Although programming a Windows application is somewhat complex, the good news is that all of the difficult code has already been written, and is available for study and re-use. The details on establishing the communi- cation link with Zemax have already been worked out, and the sample code may be "cut and paste" with only the code relating to the specific computation required needing change.
There is also a C source code file, called ZCLIENT, that vastly simplifies the DDE communication with Zemax by encapsulating all the communication in wrapper functions. ZCLIENT provides a number of functions that can be called to extract data from Zemax without requiring knowledge of DDE programming. ZCLIENT is described later in this chapter. There are complete examples of client programs and Extensions, including source code, provided with Zemax and discussed at the end of this chapter.
The power of the Extension comes at a price, although it is a reasonable one. Use of the DDE features does require that the user have a suitable compiler or development tool that can generate 32 bit Windows executables. It is also assumed that the user can write the required code, and most importantly, ensure that the code is reliable and bug-free. To maximize speed, Zemax performs very little error checking on data returned by the client, and so buggy Extensions are capable of bringing Zemax to a crash, or placing Zemax in an infinite loop.
For this reason, technical support on the implementation of DDE or Extensions is strictly limited to illustrating that the provided sample files work correctly. If you need a Zemax Extension, and do not possess the desire or ability to write them yourself, please feel free to contact Zemax Technical Support for a quote on developing a custom program to meet your requirements. We have considerable experience in developing these types of programs, and can generally write Extensions at very competitive rates on short notice.

Establishing the link

To establish a DDE link with Zemax, the client program must broadcast a message to all top level windows that includes a reference to the application name, and the topic name. For Zemax, the application name is "Zemax", and the topic name can be any non-NULL string. If two copies of Zemax are running, the application name may also be either "Zemax1" or "Zemax2" to distinguish between the two. Zemax does not use the topic name, just the application name and the "item". The item indicates to Zemax what data is being requested. The various items Zemax supports are described later.

See the many sample DDE client codes for specific code examples.

Once the DDE link is established, any analysis that can be done based upon ray tracing may be performed by the server, and the data returned to the client.

Comments on the DDE timeout interval

DDE is based upon message-passing. To prevent either the client or the server from waiting forever for a response to a message that may never come, a maximum wait time must be specified. This wait time is specified in the client application. In the sample interface code ZCLIENT, the default wait time is defined by the source code line

```
#define DDE_TIMEOUT 50000
```

This `DDE_TIMEOUT` is in milliseconds, so 50000 ms is 50 seconds. It is critical that no DDE item command requires longer than this length of time to complete. If the computation requested exceeds this time, the message passing will fail and invalid data may result. The default time limit may be edited to a larger value, however, more care must be taken in programming reliable code as the client and server applications may appear to "hang" if the message passing is not done correctly.

Terminating the link

When all computations are complete, and the DDE link is to be terminated by the client, the client sends a termination message to the Zemax server. This message will terminate only the server portion of Zemax, not Zemax itself.

Extracting data from Zemax

Zemax supports a number of capabilities under DDE. Each function is given a name, called the "item", and this item is passed to the Zemax server using the WM_DDE_REQUEST message. Any data Zemax requires (for example to trace a ray) is passed to Zemax by encoding within the item name. Most items in Zemax have names that require no further encoding, but some items have coded values appended to the name. The values are appended to the item name and are delimited by commas.

Zemax responds to each item request with a WM_DDE_DATA message, and the message contains a pointer to a block of memory holding a string. Most data is passed from Zemax back to the client application in a string, this corresponds to CF_TEXT format. The client application must extract the string and free the memory for the string.

See the many sample DDE client codes for specific code examples.

The data items

Here are the supported data items:

CloseUDOData

This item is used to close the user defined operand buffer, which allows the Zemax optimizer to proceed. See "SetUDOItem" for details. The syntax for CloseUDOData is:

```
CloseUDOData, buffercode
```

See also GetUDOSystem and SetUDOItem.

DeleteConfig

This item deletes an existing configuration in the multi-configuration editor. The syntax is:

```
DeleteConfig, config
```

See also InsertConfig.

DeleteMCO

This item deletes a multi-configuration operand in the multi-configuration editor. The syntax is:

```
DeleteMCO, operand
```

The operand argument is an integer between 1 and the current number of operands, inclusive. The return value is the new number of operands. See also InsertMCO.

DeleteMFO

This item deletes an optimization operand in the merit function editor. The syntax is:

```
DeleteMFO, operand
```

The operand argument is an integer between 1 and the current number of operands, inclusive. The return value is the new number of operands. See also InsertMFO.

DeleteObject

This item deletes an existing object. The syntax is:

```
DeleteObject,surf,object
```

See also InsertObject.

DeleteSurface

This item deletes an existing surface. The syntax is:

```
DeleteSurface,surf
```

See also InsertSurface.

ExportCAD

This item export lens data in IGES/STEP/SAT format. The syntax is:

```
ExportCAD,filename,filetype,numspline,first,last,rayslayer,lenslayer,ex
portdummy,
usesolids,raypattern,numrays,wave,field,deletevignetted,dummythick,spli
t,scatter, usepol,config
```

For details on this feature and the meaning of all parameters, see "EXPORTCAD", and "Export CAD File".

There is a complexity in using this feature via DDE. The export of lens data may take a long time relative to the timeout interval of the DDE communication. Therefore, calling this data item will cause Zemax to launch an independent thread to process the request. Once the thread is launched, the return value is the string "Exporting filename". However, the actual file may take many seconds or minutes to be ready to use. To verify that the export is complete and the file is ready, use the ExportCheck data item. ExportCheck will return 1 if the export is still running, or 0 if it has completed. Generally, the ExportCheck data item call will need to be placed in a loop which executes until ExportCheck returns 0. A typical loop test in C code might look like this:

```
/* check if the export is done */
still_working = 1;
while(still_working)
      {
      // Delay for 200 milliseconds
      Sleep(200);
      PostRequestMessage("ExportCheck", szBuffer);
      if (szBuffer[0] == '1')
            {
            // Still working!
            }
      else
            {
            // All done!
            still_working = 0;
            }
      }
```

If ExportCad returns "BUSY!" then the export thread is already running from a previous request, and the export will not be performed.

ExportCheck

This item indicates the status of the last executed ExportCAD data item. The return value is 1 if still running, or 0 if it has completed. See ExportCad above for a discussion of the use of this data item.

FindLabel

This item finds the surface that has the integer label associated with the specified surface. The syntax is
```
FindLabel,label
```
The returned value is the surface number of the first surface with the identical integer label, or -1 if no surface has the specified label. See also SetLabel and GetLabel.

GetAddress

This item extracts the address line 1, 2, 3 or 4 from the environment settings. The syntax is
```
GetAddress,1
```
with similar commands for lines 2, 3, and 4. If the user has selected to hide the address box, then all strings return as "\r\n".

GetAperture

This item extracts surface aperture data. The syntax is
```
GetAperture,surf
```
The returned string is formatted as follows:
"type, min, max, xdecenter, ydecenter, aperturefilename"
This item returns the type as an integer code; 0 for no aperture, 1 for circular aperture, 2 for circular obscuration, 3 for spider, 4 for rectangular aperture, 5 for rectangular obscuration, 6 for elliptical aperture, 7 for elliptical obscuration, 8 for user defined aperture, 9 for user defined obscuration, and 10 for floating aperture. The min and max values have different meanings for the elliptical, rectangular, and spider apertures than for circular apertures; see "Aperture type and other aperture controls" for details. See also SetAperture.

GetApodization

This item computes the intensity apodization of a ray from the apodization type and value. The syntax is:
```
GetApodization, px, py
```
Where px and py are the normalized pupil coordinates. The returned string is the intensity apodization.

GetAspect

This item extracts the graphic display aspect ratio and the width or height of the printed page in current lens units. For example, If the current aspect ratio is 3 x 4, the aspect ratio returned will be 0.75. Knowing the correct aspect ratio is required when drawing isometric plots. The format of the returned data is aspect, width. If the aspect ratio is greater than 1, then the plot is taller than it is wide, and the format of returned data is aspect, height.
The syntax is:
```
GetAspect,filename
```
where filename is the name of the temporary file associated with the window being created or updated. If the temporary file name is left off; then the default aspect ratio and width (or height) is returned.

GetBuffer

The GetBuffer item is used to retrieve client specific data from a window being updated. The syntax is:
```
GetBuffer,n,tempfile
```
where n is the buffer number, which must be between 0 and 15 inclusive; and tempfile is the name of the temporary file for the window being updated. The tempfile name is passed to the client when Zemax calls the client; see the discussion "How Zemax calls the client" for details. Note each window may have it's own buffer data, and Zemax uses the filename to identify the window for which the buffer contents are required. See also SetBuffer.

GetComment

This item extracts the surface comment, if any. The surface for which the comment data is returned is appended to the item name, for example, to get the comment for surface 5, the item name should be "GetComment,5".

GetConfig

This item extracts the current configuration number, the number of configurations, and the number of multiple configuration operands.

The returned string is formatted as follows: "currentconfig, numberconfig, numbermcoper" See also SetConfig.

GetDate

This item extracts the current date and time, in the format selected by the user in the Zemax environment dialog box.

GetExtra

This item extracts the extra surface data. The syntax is

```
GetExtra, surf, column
```

The returned string is formatted as follows: "value".

See also SetExtra.

GetField

This item extracts field data. The syntax is

```
GetField, n
```

where n is zero or the field number. If n is zero, the returned string is formatted as follows: "type, number, max_x_field, max_y_field, normalization_method"

The parameter type is an integer; either 0, 1, 2, or 3 for angles in degrees, object height, paraxial image height, or real image height, respectively. The parameter number is the number of fields currently defined. The max_x_field and max_y_field values are used to normalize field coordinates, see "Normalized field coordinates". The field normalization method is 0 for radial and 1 for rectangular.

If n is not zero, but corresponds to a valid field number, the returned string is formatted as follows: "xfield, yfield, weight, vdx, vdy, vcx, vcy, van"

These 8 values are exponential format floating point numbers for the various field data. See also SetField.

GetFile

This item extracts the full name of the lens, including the drive and path. Extreme caution should be used if the file is to be tampered with; since at any time Zemax may read or write from/to this file.

GetFirst

This item extracts first order data about the lens. The returned string is formatted as follows: "focal, pwfn, rwfn, pima, pmag"

Values in the string are the EFL, paraxial working F/#, real working F/#, paraxial image height, and paraxial magnification.

GetGlass

This item extracts some data about the glass on any surface. The syntax is

```
GetGlass,surf
```

If the specified surface is not valid, is not made of glass, or is gradient index, the returned string is empty. This data may be meaningless for glasses defined only outside of the FdC band. Otherwise, the returned string is formatted as follows: "name, nd, vd, dpgf"

GetGlobalMatrix

This item returns the matrix required to convert any local coordinates (such as from a ray trace) into global coordinates. For details on the global coordinate matrix, see "Global Coordinate Reference Surface".

The syntax for this item is:

```
GetGlobalMatrix,surf
```

The returned data string is in the format

"R11,R12,R13,R21,R22,R23,R31,R32,R33,Xo,Yo,Zo"

GetIndex

This item extracts index of refraction data for any surface. The syntax is

    GetIndex,surf

If the specified surface is not valid, or is gradient index, the returned string is empty. Otherwise, the returned string is formatted as follows:

"n1, n2, n3..."

where the index values correspond to the index at each defined wavelength, in order.

GetLabel

This item retrieves the integer label associated with the specified surface. The syntax is

    GetLabel,surf

The returned value is the label. See also SetLabel and FindLabel.

GetMulticon

This item extracts data from the multi-configuration editor. The syntax is

    GetMulticon, config, row

If the config number is greater than zero and less than or equal to the number of configurations, the returned string is formatted as follows:

"value, num_config, num_row, status, pickuprow, pickupconfig, scale, offset"

If the config number is zero, the multicon operand type data is returned as follows: "operand_type, number1, number2, number3"

The numbers are used as described in "Summary of multi-configuration operands". The status integer is 0 for fixed, 1 for variable, 2 for pickup, and 3 for thermal pickup. If status is 2 or 3, the pickuprow and pickupconfig values indicate the source data for the pickup solve.

See also SetMulticon.

GetName

This item extracts the name of the lens. The returned string is the current lens name as entered in the Title/Notes section of the System Explorer.

GetNSCData

This item gets the various data for NSC groups. The syntax is:

    GetNSCData, surf, code

Surf refers to the surface number of the NSC group, use 1 if the program mode is Non-Sequential. Currently only code = 0 is supported, in which case the returned data is the number of objects in the NSC group.

GetNSCMatrix

This item returns the rotation and position matrices for NSC objects relative to the NSC surface origin. The syntax for this item is:

    GetNSCMatrix,surf, object

The returned data string is in the format

"R11,R12,R13,R21,R22,R23,R31,R32,R33,Xo,Yo,Zo"

GetNSCObjectData

This item gets the various data for NSC objects. The syntax is:

    GetNSCObjectData, surf, object, code

Surf and object refer to the surface number and object number. The code is one of the integer values in the table below; the returned value is the data item indicated. See also SetNSCObjectData.

Code	Data returned by GetNSCObjectData
0	Object type name. (string)

1	Comment, which also defines the file name if the object is defined by a file. (string)
2	Color. (integer)
5	Reference object number. (integer)
6	Inside of object number. (integer)
The following codes set values on the Type tab of the Object Properties dialog.	
3	1 if object uses a user defined aperture file, 0 otherwise. (integer)
4	User defined aperture file name, if any. (string)
29	Sets the "Use Pixel Interpolation" checkbox. Use 1 for checked, 0 for unchecked.
The following codes set values on the Sources tab of the Object Properties dialog.	
101	Sets the source object random polarization. Use 1 for checked, 0 for unchecked.
102	Sets the source object reverse rays option. Use 1 for checked, 0 for unchecked.
103	Sets the source object Jones X value.
104	Sets the source object Jones Y value.
105	Sets the source object Phase X value.
106	Sets the source object Phase Y value.
107	Sets the source object initial phase in degrees value.
108	Sets the source object coherence length value.
109	Sets the source object pre-propagation value.
110	Sets the source object sampling method; 0 for random, 1 for Sobol sampling.
111	Sets the source object bulk scatter method; 0 for many, 1 for once, 2 for never.
The following codes set values on the Bulk Scatter tab of the Object Properties dialog.	
202	Sets the Mean Path value.
203	Sets the Angle value.

211-226	Sets the DLL parameter 1-16, respectively.

GetNSCObjectFaceData

This item gets the various data for NSC object faces. The syntax is:

```
GetNSCObjectFaceData, surf, object, face, code
```

Surf and object refer to the surface number and object number. Face is the face number. The code is one of the integer values in the table below; the returned value is the data item indicated. See also SetNSCObjectFace- Data.

Code	Data returned by GetNSCObjectFaceData
10	Coating name. (string)
20	Scatter code. Use 0 for None, 1 for Lambertian, 2 for Gaussian, 3 for ABg, and 4 for user defined, 5 for BSDF, and 6 for ABg File. (integer)
21	Scatter fraction. (double)
22	Number of rays to scatter. (integer)
23	Gaussian scatter sigma. (double)
24	Face is setting; use 0 for object default, 1 for reflective, 2 for absorbing. (integer)
30	ABg scatter profile name for reflection. (string)
31	ABg scatter profile name for transmission. (string)
40	User Defined Scatter DLL name. (string)
41-46	User Defined Scatter Parameter 1 - 6. (double)
50	BSDF file name for reflection. (string)
51	BSDF file name for transmission. (string)
52	ABg File data file name for reflection. (string)
53	ABg File data file name for transmission. (string)
54	IS Scatter Catalog data file name for reflection. (string)
55	IS Scatter Catalog data file name for transmission. (string)
56	Thin Window Scattering setting; use 0 for off, 1 for on. (integer)
60	User Defined Scatter data file name. (string)

GetNSCParameter

This item gets the parameter data for NSC objects. The syntax is:

```
GetNSCParameter, surf, object, parameter
```

Surf and object refer to the surface number and object number, and parameter is the integer parameter number. The returned string is the parameter value. See also SetNSCParameter.

GetNSCPosition

This item gets the position data for NSC objects. The syntax is:

```
GetNSCPosition, surf, object
```

Surf and object refer to the surface number and object number. The returned string is
"x, y, z, tilt-x, tilt-y, tilt-z, material" See also SetNSCPosition.

GetNSCProperty

This item mimics the ZPL function NPRO (see "NPRO(surf, object, code, face)"). The syntax is:

```
GetNSCProperty, surface, object, code, face
```

Surface and object refer to the surface number and object number, and code refers to the integer code used to identify specific types of NSC object date as described in "SETNSCPROPERTY". The face argument is used by some codes to identify a particular object face. GetNSCProperty can return both text and numeric data. The return string format is
"data"
where data is either the string or numeric data retrieved. See also SetNSCProperty

GetNSCSettings

This item extracts the maximum number of intersections, segments, nesting level, minimum absolute intensity, minimum relative intensity, glue distance, miss ray distance, and ignore errors flag used for NSC ray tracing. The data is returned in a string as follows:
"Max_Intersections, Max_Segments, Max_Nesting, Min_Absolute_Intensity, Min_Relative_Intensity, Glue_Distance, Miss_Ray_Length, Ignore_Errors"
Ignore_Errors is 1 if true, 0 if false. See also SetNSCSettings.

GetNSCSolve

This item extracts the current solve status and settings for NSC position and parameter data. The syntax is:

```
GetNSCSolve, surface, object, parameter
```

Surface and object refer to the surface number and object number. The parameter value is -1 through -6 to extract data for the x, y, z, tilt x, tilt y, and tilt z data, respectively. The parameter value is 1 or greater to extract data for the specified parameter. The data is returned in a string as follows:
"status, pickup_object, pickup_column, scale_factor, offset"
The status value is 0 for fixed, 1 for variable, and 2 for a pickup solve. Only when the status is a pickup solve is the other data meaningful.

GetOperand

This item gets the operand data from the Merit Function Editor. The syntax is:

```
GetOperand, row, column
```

Row is the operand row number in the Merit Function Editor. Column is 1 for the operand type as a string, such as "EFFL", 2 for int1, 3 for int2, 4-7 for data1-data4, 8 for target, 9 for weight, 10 for value, 11 for percent contribution, and 12-13 for data5-data6. The returned string is the numeric or string value of the requested parameter. To get the comment defined on a BLTH operand, set column equal to 2. To update the merit function prior to calling GetOperand, use the Optimize item with the number of cycles set to -1. See also SetOperand and Optimize.

GetPath

This item extracts the full path name to the <data> folder, and the path name to the default folder for lenses. The returned strings are separated by a comma.

GetPolState

This item extracts the default polarization state set by the user. The data is formatted as follows: "nIsPolarized, Ex, Ey, Phax, Phay"

If nIsPolarized is anything other than zero, then the default polarization state is unpolarized. Otherwise, the Ex, Ey, Phax, and Phay values are used to define the polarization state. Ex and Ey should each be normalized to a magnitude of unity, although this is not required. Phax and Phay are in degrees. See also SetPolState.

GetPolTrace

This item is very similar to GetTrace, with the added capability to trace polarized rays through the system. See the GetTrace documentation for details. The syntax for the GetPolTrace item is as follows:

    GetPolTrace, wave, mode, surf, hx, hy, px, py, Ex, Ey, Phax, Phay

The arguments are identical to GetTrace, except for the additional Ex, Ey, Phax, and Phay arguments. Ex and Ey are the normalized electric field magnitudes in the x and y directions. The quantity Ex*Ex + Ey*Ey should have a value of 1.0 (with an important exception described below) although any values are accepted. Phax and Phay are the relative phase, in degrees.

If Ex, Ey, Phax, and Phay are all zero, and only in this case, then Zemax assumes an "unpolarized" ray trace is required. An unpolarized ray trace actually requires Zemax to trace two orthogonal rays and the resulting transmitted intensity be averaged. If any of the four values are not zero, then a single polarized ray will be traced.

For example, to trace the real unpolarized marginal ray to the image surface at wavelength 2, the item string would be

    GetPolTrace,2,0,-1,0.0,0.0,0.0,1.0,0,0,0,0

For polarized rays, the data message comes back in the following format: "error, intensity, Exr, Eyr, Ezr, Exi, Eyi, Ezi"

The integer error will be zero if the ray traced successfully, otherwise it will be a positive or negative number. If positive, then the ray missed the surface number indicated by error. If negative, then the ray total internal reflected (TIR) at the surface given by the absolute value of the error number. Always check to verify the ray data is valid before using the rest of the string!

The intensity will be the transmitted intensity. It is always normalized to an input electric field intensity of unity. The transmitted intensity accounts for surface, thin film, and bulk absorption effects, but does not consider whether or not the ray was vignetted. The Ex, Ey, and Ez values are the electric field components, with the r and i characters denoting the real and imaginary portions.

For unpolarized rays, the data message is: "error, intensity"

Although GetPolTrace is easy to program and use, there is a significant disadvantage to using GetPolTrace: only one ray per DDE call is traced. The overhead of passing data via DDE can be large compared to tracing the ray, and so if a large number of rays are required to be traced, the execution may be relatively slow. For information on tracing large numbers of rays, see the discussion "Tracing large numbers of rays" elsewhere in this chapter. See also GetPolTraceDirect.

GetPolTraceDirect

GetPolTraceDirect provides the same direct access to the Zemax ray tracing engine that GetTraceDirect does, and allows for polarized ray tracing as well. See GetTraceDirect for important details on the direct access version of GetTrace. The syntax for the GetPolTraceDirect item is:

GetPolTraceDirect, wave, mode, startsurf, stopsurf, x, y, z, l, m, n, Ex, Ey, Phax, Phay

The Ex, Ey, Phax, and Phay parameters are exactly the same as defined for GetPolTrace. The data message comes back in the same format as described in GetPolTrace.

GetPupil

This item extracts pupil data. The returned string is formatted as follows:
"type, value, ENPD, ENPP, EXPD, EXPP, apodization_type, apodization_factor"

The parameter type is an integer indicating the system aperture type, a number between 0 and 5, for entrance pupil diameter, image space F/#, object space NA, float by stop size, paraxial working F/#, or object cone angle, respectively. The value is the system aperture value, unless float by stop size is being used, in which case the value is the stop surface semi-diameter. The next 4 values are the entrance pupil diameter, entrance pupil position, exit pupil diameter, and exit pupil position, all in lens units. The apodization_type is an integer which is set to 0 for none, 1 for Gaussian, 2 for Tangential. The apodization_factor is the "Apodization Factor" shown in the Aperture section of the System Explorer, and is only available when Gaussian Apodization is selected.

GetRefresh

This data item causes Zemax to copy the all lens data (wavelengths, fields, editors, etc.) into the stored copy of the server. The lens is then updated, which means Zemax recomputes all pupil positions, solves, and index data. If the lens can be updated, Zemax returns the string "0", otherwise, it returns "-1". If the GetUpdate returns "-1", no ray tracing can be performed.

All subsequent commands will now affect or be executed upon the newly copied lens data. The old lens data, if any, cannot be recovered. See also GetUpdate and PushLens.

GetSag

This item computes the sag of any surface. The item name is formatted as "GetSag,surf,x,y", where surf is the surface number, and x and y are the coordinates on the surface for which the sag is computed. The returned string is formatted as "sag, alternatesag". The values for x, y, and the sag are all in lens units.

GetSequence

This item returns the sequence number of the lens in the Server's memory, then the sequence number of the lens in the LDE, separated by a comma.

GetSerial

This item returns the serial number of the Zemax key.

GetSettingsData

This item retrieves the settings data used by a window. The data must have been previously stored by a call to SetSettingsData. The data may have been stored by a previous execution of the client program. The syntax is:

```
GetSettingsData, tempfile, number
```

The tempfile data is the name of the output file passed by Zemax to the client. Zemax uses this name to identify for which window the GetSettingsData request is for.

The number is the data number used by the previous SetSettingsData call. Currently, only number = 0 is supported. This number may be used to expand the feature in the future.

The return value is the string that was saved by a previous SetSettingsData command for the window and number. See also "How Zemax calls the client" and SetSettingsData.

GetSolve

This item returns data about any solve on any surface. The syntax is "GetSolve,surface,code" where code is an integer code indicating which surface parameter the solve data is for. The solve data is returned in the following formats, depending upon the code value.

GetSolve Code	Returned data format
0, curvature	solvetype, parameter1, parameter2, pickupcolumn
1, thickness	solvetype, parameter1, parameter2, parameter3, pickupcolumn
2, glass	solvetype (for solvetype = 0)
	solvetype, Index, Abbe, Dpgf (for solvetype = 1, model glass)
	solvetype, pickupsurf (for solvetype = 2, pickup)
	solvetype, index_offset, abbe_offset (for solvetype = 4, offset)
	solvetype (for solvetype = all other values)
3, semi-diameter	solvetype, pickupsurf, pickupcolumn

4, conic	solvetype, pickupsurf, pickupcolumn
5-16, parameters 1-12	solvetype, pickupsurf, offset, scalefactor, pickupcolumn
17, parameter 0	solvetype, pickupsurf, offset, scalefactor, pickupcolumn
1001+, extra data values 1+	solvetype, pickupsurf, scalefactor, offset, pickupcolumn

The solvetype is an integer code, and the parameters have meanings that depend upon the solve type; see the chapter "SOLVES" for details. See also SetSolve.

GetSurfaceData

This item gets surface data. The syntax is:
```
GetSurfaceData, surf, code
```
or
```
GetSurfaceData, surf, code, arg2
```
Surf refers to the surface number. The code is one of the integer values in the table below; the returned value is the data item indicated. See also SetSurfaceData. Not all codes use the arg2 value.

Code	Date returned by GetSurfaceData
0	Surface type name. (string)
1	Comment. (string)
2	Curvature (not radius).
3	Thickness.
4	Glass. (string)
5	Semi-Diameter.
6	Conic.
7	Coating. (string)
8	Thermal Coefficient of Expansion (TCE).
9	User Defined Surface DLL name. (string)
20	Ignore surface flag, 0 for not ignored, 1 for ignored.
51	Before tilt and decenter order; 0 for Decenter then Tilt, 1 for Tilt then Decenter.
52-56	Before decenter x, decenter y, tilt x, tilt y, and tilt z values, respectively.

60	After status; 0 for explicit, 1 for pickup this surface, 2 for reverse this surface, 3 for pickup this surface - 1, 4 for reverse this surface - 1, etc.
61	After tilt and decenter order; 0 for Decenter then Tilt, 1 for Tilt then Decenter.
62-66	After decenter x, decenter y, tilt x, tilt y, and tilt z values, respectively.
70	Use Layer Multipliers and Index Offsets. Use 1 for true, 0 for false.
71	Layer Multiplier value. The coating layer number is defined by arg2.
72	Layer Multiplier status. Use 0 for fixed, 1 for variable, or n+1 for pickup from layer n. The coating layer number is defined by arg2.
73	Layer Index Offset value. The coating layer number is defined by arg2.
74	Layer Index Offset status. Use 0 for fixed, 1 for variable, or n+1 for pickup from layer n. The coating layer number is defined by arg2.
75	Layer Extinction Offset value. The coating layer number is defined by arg2.
76	Layer Extinction Offset status. Use 0 for fixed, 1 for variable, or n+1 for pickup from layer n. The coating layer number is defined by arg2.
80	Coordinate return solve status. Use 0 for none, 1 for Orientation only, 2 for Orientation+XY, and 3 for Orientation + XYZ. The surface must be a coordinate break for this solve to have any affect.
81	Coordinate return surface number. Used only by the coordinate return solve.

For example, to get the conic data for surface 8, the item name should be "GetSurfaceData,8,6". To get the coating multiplier for layer 4 on surface 7, the item name should be "GetSurfaceData,7,71,4".
See also SetSurfaceData and GetSurfaceParameter.

GetSurfaceDLL
This item extracts the name of the DLL if the surface is a user defined type. The surface for which the surface DLL name is returned is appended to the item name, for example, to get the DLL name for a user defined surface 5, the item name should be "GetSurfaceDLL,5".
The returned string is formatted as follows: "dllname, surfacename"
The dllname is the name of the defining DLL, the surfacename is the string displayed by the DLL in the surface type column of the Lens Data Editor.

GetSurfaceParameter
This item extracts surface parameter data. The surface and parameter number to be returned is appended to the item name, for example, to get the parameter 5 data for surface 8, the item name should be "GetSurface- Parameter,8,5".
See also GetSurfaceData and SetSurfaceParameter.

GetSystem
This item extracts system data.
The returned string is formatted as follows:
"numsurfs, unitcode, stopsurf, nonaxialflag, rayaimingtype, adjust_index, temp, pressure, globalrefsurf"

This item returns the number of surfaces, the unit code (0, 1, 2, or 3 for mm, cm, in, or M), the stop surf number, a flag to indicate if the system is non-axial symmetric (0 for axial, 1 if not axial), the ray aiming (0 for off, 1 for paraxial, 2 for real), the adjust index data to environment flag (0 if false, 1 if true), the current temperature and pressure, and the global coordinate reference surface number. The need_save value indicates whether the file has been modified
See also SetSystem and GetSystemAper.

GetSystemAper

This item extracts system aperture data. The returned string is formatted as follows: "type, stopsurf, aperture_value"
This item returns the system aperture type (0, 1, 2, 3, 4, or 5 for Entrance Pupil Diameter, Image Space F/#, Object Space NA, Float by Stop Size, Paraxial Working F/#, or Object Cone Angle, respectively), the stop surface number, and the system aperture value. If the aperture type is Float by Stop Size, the aperture value is the stop surface semi-diameter.
See also GetSystem and SetSystemAper.

GetSystemProperty

This item mimics the ZPL function SYPR (see "SYPR(code)"). The syntax is:
```
 GetSystemProperty code
```
Code is an integer value that defines the specific system property requested. For a listing of these codes, see "SETSYSTEMPROPERTY, SYSP". The return string format is
"data"
where data is either the string or numeric data retrieved. See also SetSystemProperty

GetTextFile

This item creates a text file of any Zemax analysis that supports text. The syntax is
```
 GetTextFile, "textfilename", type, "settingsfilename", flag
```
The textfilename must be in quotes and include the full path, name, and extension for the file to be created.
The type argument is a 3 character case-sensitive label that indicates the type of analysis to be performed. For the full list of string codes, see the "String Codes" section of the The Programming Tab. The labels are case sensitive. If no label is provided or recognized, a standard ray trace will be generated.
If a valid file name is used for the "settingsfilename" argument, Zemax will use or save the settings used to compute the text file, depending upon the value of the flag parameter.
If the flag value is 0, then the default settings will be used. If the lens file has it's own default settings, then those will be used; these are the settings stored in the "lensfilename.cfg" file. If no lens specific default settings exist, then the default settings for all Zemax files, stored in the file "Zemax.CFG" will be used, if any. If no previous settings have been saved for this or any other lens, then the default settings used are the "factory" defaults used by Zemax.
If the flag value is 1, then the settings provided in the settings file, if valid, will be used to generate the file. If the data in the settings file is in anyway invalid, then the default settings will be used to generate the file. The only valid settings files are those generated by Zemax, then renamed and saved to a new user defined file name. For example, on the Spot Diagram settings dialog, pressing "Save" will generate a "SPT.CFG" file with the saved settings. If this file is renamed to "MySPT.CFG", then the file, with a full path, may be used as the "settingsfile- name" argument to GetTextFile.
If the flag value is 2, then the settings provided in the settings file, if valid, will be used and the settings box for the requested feature will be displayed. After the user makes any changes to the settings the file will then be generated using the new settings. The dialog boxes used to change the analysis settings use the data from the lens currently in the Lens Data Editor, so the item PushLens must be used prior to using flag = 2.
No matter what the flag value is, if a valid file name is provided for the settingsfilename, the settings used will be written to the settings file, overwriting any data in the file.
Only text, and not graphic files, are supported by GetTextFile. See also OpenWindow.

GetTol

This item extracts tolerance data. The syntax is
```
 GetTol, n
```
where n is zero or the tolerance operand number. If n is zero, the returned string is formatted as follows: "number"

where number is the number of tolerance operands defined. If n is not zero, but corresponds to a valid tolerance operand number, the returned string is formatted as follows:
"toltype, int1, int2, min, max, int3"

GetTrace

This item requires that the client provide additional data. In order to trace a ray, Zemax needs to know the relative field and pupil coordinates, the wavelength, the mode (either real, mode = 0 or paraxial, mode = 1) as well as the surface to trace the ray to. All of this data is encoded and appended to the item name. This is accomplished by delimiting the various parameters with commas as follows:

`GetTrace, wave, mode, surf, hx, hy, px, py`

For example, to trace the real chief ray to surface 5 at wavelength 3, the item string would be

`GetTrace,3,0,5,0.0,1.0,0.0,0.0`

Although this looks cumbersome, it is easy to program. Zemax receives the item, recognizes the "GetTrace" at the beginning of the name, strips off the rest of the string, then parses it, and traces the ray. Usually, the ray data is only needed at the image surface; setting the surface number to -1 will yield data at the image surface. Note 0 is reserved for the object surface.

The data message comes back in the following format: "error, vigcode, x, y, z, l, m, n, l2, m2, n2, intensity"

The integer error will be zero if the ray traced successfully, otherwise it will be a positive or negative number. If positive, then the ray missed the surface number indicated by error. If negative, then the ray total internal reflected (TIR) at the surface given by the absolute value of the error number. Always check to verify the ray data is valid before using the rest of the string!

The parameter vigcode is the first surface where the ray was vignetted. Unless an error occurs at that surface or subsequent to that surface, the ray will continue to trace to the requested surface.

The x, y, and z values are the coordinates on the requested surface.

The l, m, and n values are the direction cosines after refraction into the media following the requested surface. The l2, m2, and n2 values are the surface intercept direction normals at the requested surface.

The intensity is the relative transmitted intensity of the ray, including any pupil or surface apodization defined.

Although GetTrace is easy to program and use, there is a significant disadvantage to using GetTrace: only one ray per DDE call is traced. The overhead of passing data via DDE can be quite large compared to tracing the ray, and so if a large number of rays are required to be traced, the execution may be relatively slow. For information on tracing large numbers of rays, see the discussion "Tracing large numbers of rays" elsewhere in this chapter. See also GetTraceDirect.

GetTraceDirect

GetTraceDirect provides a more direct access to the Zemax ray tracing engine. Normally, rays are defined by the normalized field and pupil coordinates hx, hy, px, and py. Zemax takes these normalized coordinates and computes the object coordinates (x, y, and z) and the direction cosines to the entrance pupil aim point (l, m, and n; for the x-, y-, and z-direction cosines, respectively).

However, there are times when it is more appropriate to trace rays by direct specification of x, y, z, l, m, and n. The direct specification has the added flexibility of defining the starting surface for the ray anywhere in the optical system. This flexibility comes at the price of requiring the client program to carefully ensure that the starting ray coordinates are meaningful.

Like GetTrace, this item requires that the client provide additional data. In order to trace a ray, Zemax needs to know x, y, z, l, m, n, the wavelength, the mode (either real, mode = 0 or paraxial, mode = 1) as well as the starting and stopping surfaces to trace the ray from and to. All of this data is encoded and appended to the item name. This is accomplished by delimiting the various parameters with commas as follows:

`GetTraceDirect, wave, mode, startsurf, stopsurf, x, y, z, l, m, n`

The data message comes back in the following format: "error, vigcode, x, y, z, l, m, n, l2, m2, n2, intensity" where the parameters are exactly the same as for GetTrace, except for intensity.The intensity is the relative transmitted intensity of the ray, excluding any pupil apodization defined. Note GetTrace includes pupil apodiza- tion, GetTraceDirect does not. Both include surface apodizations.

Although GetTraceDirect is easy to program and use, there is a significant disadvantage to using GetTrace- Direct: only one ray per DDE call is traced. The overhead of passing data via DDE can be quite large compared to tracing the ray, and so if a large number of rays are required to be traced, the execution may be relatively slow. For information on tracing large numbers of rays, see the discussion "Tracing large numbers of rays" elsewhere in this chapter. See also GetTrace.

GetUDOSystem

This item is used to load a particular lens from the optimization function memory into the Zemax server's memory. The only time this item name should be used is when implementing a User Defined Operand, or UDO. UDO's are described in detail in the chapter "Optimization". See that chapter for details. The syntax for GetUDOSystem is

```
GetUDOSystem, buffercode
```

where buffercode is an integer passed to the UDO on the command line. See also SetUDOData.

GetUpdate

This data item causes Zemax to update the lens, which means Zemax recomputes all pupil positions, solves, and index data. If the lens can be updated, Zemax returns the string "0", otherwise, it returns "-1". If the GetUpdate returns "-1", no ray tracing can be performed. To update the merit function, use the Optimize item with the number of cycles set to -1. See also GetRefresh and Optimize.

GetVersion

This item returns the version number of Zemax. This number is generally a 5 digit integer with no decimal place, such as 10000 for version 10.0. Later versions will have larger numbers.

GetWave

This item extracts wavelength data. The syntax is

```
GetWave, n
```

where n is zero or the wavelength number. If n is zero, the returned string is formatted as follows: "primary, number"
The parameter primary is an integer which indicates which of the wavelengths is the primary one. The parameter number is the number of wavelengths currently defined. If n is not zero, but corresponds to a valid wavelength number, the returned string is formatted as follows:
"wavelength, weight"
These 2 values are exponential format floating point numbers which correspond to the value and weight for the specified wavelength. See also SetWave.

Hammer

Hammer calls the Zemax Hammer optimizer. For information on Hammer, see "The Hammer Optimization algorithm". The syntax is

```
Hammer, n, algorithm
```

where n is the number of cycles to run, and algorithm is 0 for damped least squares and 1 for orthogonal descent. The return value is the final merit function. If the merit function value returned is 9.0E+009, the optimization failed, usually because the lens or merit function could not be evaluated. If n is less than 1, Hammer updates all operands in the merit function and returns the current merit function, and no optimization is performed. The number of cycles should be kept small enough to allow the algorithm to complete and return before the DDE communication times out, or an error will occur.

ImportExtraData

This item imports extra data and grid surface data values into an existing surface. The syntax is:

```
ImportExtraData,surf,filename
```

For details on how Zemax imports extra data, see "Extra Data". For information on importing grid surface data, see "Importing grid data".

InsertConfig

This item inserts a new configuration in the multi-configuration editor. The syntax is:

```
InsertConfig, config
```

The new configuration will be placed at the location indicated by the parameter config. See also DeleteConfig.

InsertMCO

This item inserts a new multi-configuration operand in the multi-configuration editor. The syntax is:

```
InsertMCO, operand
```

The operand argument is an integer between 1 and the current number of operands plus 1, inclusive. The return value is the new number of operands. See also DeleteMCO.

InsertMFO

This item inserts a new optimization operand in the merit function editor. The syntax is:

```
InsertMFO, operand
```

The operand argument is an integer between 1 and the current number of operands plus 1, inclusive. The return value is the new number of operands. See also DeleteMFO.

InsertObject

This item inserts a new NSC object. The syntax is:

```
InsertObject,surf,object
```

The new null object will be placed at the location indicated by the parameters surf and object. See also SetNSCObjectData to define data for the new surface and the DeleteObject item.

InsertSurface

This item inserts a new surface. The syntax is:

```
InsertSurface,surf
```

The new surface will be placed at the location indicated by the parameter surf. See also SetSurface to define data for the new surface and the DeleteSurface item.

LoadDetector

Loads the data saved in a file to an NSC Detector Rectangle, Detector Color, Detector Polar, or Detector Volume object. The syntax is:

```
LoadDetector,surf,object,filename
```

Surf is the surface number of the non-sequential group; use 1 when using non-sequential mode. Object is the object number of the detector object. The filename may include the full path, if no path is provided the path of the current lens file is used. The extension should be DDR, DDC, DDP, or DDV for Detector Rectangle, Color, Polar, and Volume objects, respectively. The return value is zero if the load was successful, otherwise an error code is returned.

LoadFile

Loads a Zemax file into the server. The syntax is:

```
LoadFile,filename,append
```

Note that loading a file does not change the data displayed in the LDE; the server process has a separate copy of the lens data. The file name to be loaded is appended to the LoadFile item name, and must include the full path. For example: "LoadFile,C:\Zemax\SAMPLES\COOKE.ZMX".

The value append may be omitted. If present and greater than zero, then the new file is appended to the current file starting at the surface number defined by the value append.

The returned string is the same as for the GetUpdate item; after updating the newly loaded lens file. If a value other than 0 is returned, the Update failed, if -999 is returned, the file could not be loaded. See also GetPath, SaveFile, and PushLens.

LoadMerit

Loads a Zemax .MF or .ZMX file and extracts the merit function and places it in the lens loaded in the server. The syntax is

```
LoadMerit, filename
```

Note that loading a merit function file does not change the data displayed in the LDE; the server process has a separate copy of the lens data. The file name to be loaded is appended to the LoadMerit item name, and must include the full path. For example: "LoadMerit,C:\Zemax\SAMPLES\MyMerit.MF". The returned string is formatted as follows:

"number, merit"

where number is the number of operands in the merit function and merit is the value of the merit function. If the merit function value is 9.00e+009, the merit function cannot be evaluated. See also Optimize, SaveMerit.

LoadTolerance

Loads a tolerance file previously saved with SaveTolerance places the tolerances in the lens loaded in the server. The syntax is:

```
LoadTolerance, filename
```

If no path is provided in the filename, the <data>\Tolerance folder is assumed. The returned string is formatted as follows:

"number"

where number is the number of tolerance operands loaded.

MakeGraphicWindow

This item notifies Zemax that graphic data has been written to a file and may now be displayed as a Zemax child window. The primary purpose of this item is to implement user defined features in a client application, that look and act like native Zemax features. The item string must be formatted as follows:

```
MakeGraphicWindow, filename, modulename, wintitle, textflag,
settingsdata
```

or

```
SetSettingsData, 0, settingsdata
MakeGraphicWindow, filename, modulename, wintitle, textflag
```

The filename is the full path and file name to the temporary file that holds the graphic data. This must be the same name as passed to the client executable in the command line arguments, if any. The modulename is the full path and executable name of the client program that created the graphic data. The wintitle is the string which defines the title Zemax should place in the top bar of the window.

The textflag should be 1 if the client can also generate a text version of the data. Since the current data is a graphic display (it must be if the item is MakeGraphicWindow) Zemax wants to know if the "Text" menu option should be available to the user, or if it should be grayed out. If the text flag is 0, Zemax will gray out the "Text" menu option and will not attempt to ask the client to generate a text version of the data.

The settings data is a string of values delimited by spaces (not commas) which are used by the client to define how the data was generated. These values are only used by the client, not by Zemax. The settings data string holds the options and data that would normally appear in a Zemax "settings" style dialog box. The settings data should be used to recreate the data if required. Because the total length of a data item cannot exceed 255 characters, the data item SetSettingsData may be used prior to the call to MakeGraphicWindow to specify the settings data string rather than including the data as part of MakeGraphicWindow. See "How Zemax calls the client" for more details on the settings data.

A sample item string might look like this:

```
MakeGraphicWindow,C:\TEMP\ZGF001.TMP,C:\Zemax\FEATURES\CLIENT.EXE,Client
Window,1,0 1 2 12.55
```

This item indicates that Zemax should open a graphic window, display the data stored in the file C:\TEMP\ZGF001.TMP, and that any updates or setting changes can be made by calling the client module C:\Zemax\FEATURES\CLIENT.EXE. This client can generate a text version of the graphic, and the settings data string (used only by the client) is "0 1 2 12.55".

MakeTextWindow

This item notifies Zemax that text data has been written to a file and may now be displayed as a Zemax child window. The primary purpose of this item is to implement user defined features in a client application, that look and act like native Zemax features. The item string must be formatted as follows:

```
MakeTextWindow, filename, modulename, wintitle, settingsdata
```

or

```
SetSettingsData, 0, settingsdata
MakeTextWindow, filename, modulename, wintitle
```

The filename is the full path and file name to the temporary file that holds the text data. This must be the same name as passed to the client executable in the command line arguments, if any. The modulename is the full path and executable name of the client program that created the text data. The wintitle is the string which defines the title Zemax should place in the top bar of the window.

The settingsdata is a string of values delimited by spaces (not commas) which are used by the client to define how the data was generated. These values are only used by the client, not by Zemax. The settingsdata string holds the options and data that would normally appear in a Zemax "settings" style dialog box. The settingsdata should be used to recreate the

data if required. Because the total length of a data item cannot exceed 255 characters, the data item SetSettingsData may be used prior to the call to MakeTextWindow to specify the settings data string rather than including the data as part of MakeTextWindow. See "How Zemax calls the client" for more details on the settings data.

A sample item string might look like this:

```
MakeTextWindow,C:\TEMP\ZGF002.TMP,C:\Zemax\FEATURES\CLIENT.EXE,ClientWind
ow,6 5 4 12.55
```

This item indicates that Zemax should open a text window, display the data stored in the file C:\TEMP\ZGF002.TMP, and that any updates or setting changes can be made by calling the client module C:\Zemax\FEATURES\CLIENT.EXE. The settingsdata string (used only by the client) is "6 5 4 12.55".

ModifySettings

ModifySettings is used to change specific options in Zemax settings files. The settings files are used by MakeTextWindow and MakeGraphicWindow. The item string syntax is:

```
ModifySettings, filename, type, value
```

The filename is the full name of the settings file, including the path. The type is a mnemonic that defines what option value is being modified. The valid values for type are as defined in the ZPL macro command MODIFY-SETTINGS that serves the same function as ModifySettings does for extensions. See "MODIFYSETTINGS" for a complete list of the type codes. The return value is an integer code: 0 for no error, -1 for invalid file, -2 for incorrect version number, and -3 for a file access conflict.

NewLens

This item erases the current lens. The "minimum" lens that remains is identical to the lens in the Lens Data Editor when "File, New" is selected. No prompt to save the existing lens is given.

NSCCoherentData

This item returns data from an NSC detector. The syntax is

```
NSCCoherentData,surface,object,pixel,data
```

where the arguments are exactly the same as defined for the NSDC optimization operand described in "NSDC". The return value is the detector data formatted as a string.

NSCDetectorData

This item returns data from an NSC detector. The syntax is

```
NSCDetectorData,surface,object,pixel,data
```

where the arguments are exactly the same as defined for the NSDD optimization operand described in "NSDD", except that the # Ignored argument is not supported (# Ignored is always assumed to be zero). The return value is the detector data formatted as a string.

NSCLightningTrace

This item traces rays from one or all NSC sources using LightningTrace. The syntax is

```
NSCLightningTrace, surf, source, ray_sampling, edge_sampling
```

where the arguments are exactly the same as defined for the NSLT macro command described in "NSLT".

NSCTrace

This item traces rays from one or all NSC sources with various optional arguments. The syntax is

```
NSCTrace,surf,source,split,scatter,usepolar,ignore_errors,no_random_
seed,save, savefilename,filter,zrd_format
```

where the arguments are exactly the same as defined for the NSTR macro command described in "NSTR".

For information about timeout limits when executing potentially long calculations, see "Comments on the DDE timeout interval".

OpenWindow

OpenWindow will open a new analysis window on the main Zemax screen. The syntax is

```
OpenWindow,Type
```

The type argument is a 3 character case-sensitive label that indicates the type of analysis to be performed. The labels are case sensitive.

OperandValue

OperandValue returns the value of any optimization operand, even if the operand is not currently in the merit function. The syntax is

    OperandValue, type, int1, int2, data1, data2, data3, data4, data5, data6

The type argument is the name of the operand, such as EFFL. The remaining values are the usual values used by the Merit Function Editor; see "Optimization operands". Trailing zero values are not required. For example, to retrieve the EFFL at wavelength 4 the syntax is:
OperandValue, EFFL, 0, 4

Optimize

Optimize calls the Zemax Damped Least Squares optimizer. The syntax is

    Optimize, n, algorithm

where n is the number of cycles to run and algorithm is 0 for Damped Least Squares (the default) and 1 for Orthogonal Descent. For more information on these algorithms see "Performing an optimization". The return value is the final merit function. If the merit function value returned is 9.0E+009, the optimization failed, usually because the lens or merit function could not be evaluated. If n is zero, the optimization runs in automatic mode. If n is less than zero (for example, n = -1), Optimize updates all operands in the merit function and returns the current merit function, and no optimization is performed.

For information about timeout limits when executing potentially long calculations, see "Comments on the DDE timeout interval".

PushLens

The syntax is:

    PushLens, flag

PushLens will take the lens currently loaded in the server's memory and push it into the Lens Data Editor. This operation requires the permission of the user running the Zemax program. To check or set this permission status, see the PushLensPermission data item described below.

The returned string is the same as for the GetUpdate item; after updating the newly pushed lens file. If a value other than 0 is returned, the Update failed, if -999 is returned, the lens could not be pushed into the LDE.

If the flag value is zero or is omitted, the open windows are not updated. If the flag value is 1, then all open analysis windows are updated.

See also PushLensPermission, GetPath, GetRefresh, LoadFile, and SaveFile.

PushLensPermission

The syntax is:

    PushLensPermission

The return value is 1 if Zemax is set to accept PushLens commands, or 0 if extensions are not allowed to use PushLens. The PushLens data item can only succeed if the user has set the appropriate permissions in Zemax. To allow a DDE extension to use PushLens, the option "Allow Extensions To Push Lenses" must be checked. This option is found in the Editors section of the Project Preferences (which is in the Setup Tab), and is described in "Editors". The proper use of PushLens is to first call PushLensPermission. If the return value is 0, then the client should display a dialog box instructing the user to turn on the "Allow Extensions To Push Lenses" option before proceeding.

QuickFocus

This feature performs a quick best focus adjustment of the optical system. The syntax is:

    QuickFocus, mode, centroid

The value mode should be 0, 1, 2, or 3 for RMS spot radius, spot x, spot y, or wavefront OPD. The value for centroid should be 0 or 1 to indicate the RMS should be referenced to the chief ray or image centroid, respectively. The "best" focus is chosen as a wavelength weighted average over all fields. The return value is always "OK".

ReleaseWindow

When Zemax calls the client to update or change the settings used by the client function, the menu bar is grayed out on the window to prevent multiple updates or setting changes from being requested simultaneously. Normally, when the client code sends the MakeTextWindow or MakeGraphicWindow, the menu bar is once again activated. However, if during an update or setting change, the new data cannot be computed, then the window must be released. The ReleaseWindow serves just this one purpose. If the user selects "Cancel" when changing the settings, the client code should send a ReleaseWindow item to release the lock out of the menu bar. If this command is not sent, the window cannot be closed, which will prevent Zemax from terminating normally. The ReleaseWindow item takes just one argument: the name of the temporary file. The syntax is:

ReleaseWindow, filename

The return value is zero if no window is using the filename, or a positive integer number if the file is being used.

RemoveVariables

Sets all currently defined variables to fixed status. This data item takes no arguments.

SaveDetector

Saves the data currently on an NSC Detector Rectangle, Detector Color, Detector Polar, or Detector Volume object to a file. The syntax is:

 SaveDetector,surf,object,filename

Surf is the surface number of the non-sequential group; use 1 when using non-sequential mode. Object is the object number of the detector object. The filename may include the full path, if no path is provided the path of the current lens file is used. The extension should be DDR, DDC, DDP, or DDV for Detector Rectangle, Color, Polar, and Volume objects, respectively. The return value is zero if the save was successful, otherwise an error code is returned.

SaveFile

Saves the lens currently loaded in the server to a Zemax file. The file name to be used for the save is appended to the SaveFile item name, and must include the full path. For example: "SaveFile,C:\Zemax\SAM-PLES\COOKE.ZMX". The returned string is the same as for the GetUpdate item; after updating the newly saved lens file. If a value other than 0 is returned, the Update failed, if -999 is returned, the file could not be saved. See also GetPath, GetRefresh, LoadFile, and PushLens.

SaveMerit

Saves the current merit function to a Zemax .MF file. The syntax is

 SaveMerit, filename

The file name must include the full path. For example: "SaveMerit,C:\TEMP\MyMerit.MF". The returned string is formatted as follows:

"number"

where number is the number of operands in the merit function. See also Optimize, LoadMerit.

SaveTolerance

Saves the tolerances of the current lens to a file. The syntax is:

 SaveTolerance, filename

If no path is provided in the filename, the <data>\Tolerance folder is assumed. The returned string is formatted as follows:

"number"

where number is the number of tolerance operands saved.

SetAperture

This item sets surface aperture data. The syntax is

 SetAperture, surf, type, min, max, xdecenter, ydecenter, aperturefile

The returned string is formatted as follows:

"type, min, max, xdecenter, ydecenter, aperturefile"

This item uses an integer code for the surface aperture type; 0 for no aperture, 1 for circular aperture, 2 for circular obscuration, 3 for spider, 4 for rectangular aperture, 5 for rectangular obscuration, 6 for elliptical aperture, 7 for elliptical obscuration, 8 for user defined aperture, 9 for user defined obscuration, and 10 for floating aperture.

The min and max values have different meanings for the elliptical, rectangular, and spider apertures than for circular apertures; see "Aperture type and other aperture controls" for details.

If SetAperture is used to set user defined apertures or obscurations, the aperturefile must be the name of a file which lists the x, y, coordinates of the user defined aperture file in a two column format. For more information on user defined apertures, see "User defined apertures and obscurations".

See also GetAperture.

SetBuffer

The SetBuffer item is used to store client specific data with the window being created or updated. The buffer data can be used to store user selected options, instead of using the settings data on the command line of the MakeTextWindow or MakeGraphicWindow items. The data must be in a string format. The syntax is:

```
SetBuffer,1,any text you want......
```

There are 16 buffers provided, numbered 0 through 15, and each can be set using SetBuffer,0,.....; SetBuf- fer,1,.....etc. The text following the "SetBuffer,n," is the only text that is stored; and it may be a maximum of 240 characters.

Note the buffer data is not associated with any particular window until either the MakeTextWindow or MakeGraphicWindow items are issued. Once Zemax receives the MakeTextWindow or MakeGraphicWindow item, the buffer data is then copied to the appropriate window memory, and then may later be retrieved from that window's buffer using GetBuffer. See also GetBuffer.

SetConfig

This item switches the current configuration number and updates the system. The desired configuration is appended to the item name. For example, to switch to configuration 3, the item name is "SetConfig,3".

The returned string is formatted as follows: "currentconfig, numberconfig, error"

The currentconfig is the new configuration number, which will be between 1 and the value of numberconfig. Normally, this will be the desired configuration requested in the item name, as long as it was a valid configuration number. The error code is the same as returned by the GetUpdate item, and will be zero if the new current configuration is traceable. See also GetConfig.

SetExtra

This item sets the extra surface data. The syntax is

```
SetExtra, surf, column, value
```

The returned string is formatted as follows: "value"

See also GetExtra.

SetField

This item sets field data. The syntax is

```
SetField, 0, type, number, normalization
```

or

```
SetField, n, xf, yf, wgt, vdx, vdy, vcx, vcy, van
```

If the value for n is zero, then the field type, total number of fields, and field normalization type is set to the new values. The field type is 0 for angle, 1 for object height, 2 for paraxial image height, and 3 for real image height. The field normalization type is 0 for radial and 1 for rectangular.

If n is a valid field number (between 1 and the number of fields, inclusive) then the field x and y values, field weight, and vignetting factors are all set. See also GetField. The returned data is the same as for GetField, n.

SetFloat

This item sets all surfaces without surface apertures to have floating apertures. Floating apertures will vignette rays which trace beyond the semi-diameter. The syntax is

```
SetFloat
```

The returned value is "OK".

SetLabel

This item associates an integer label with the specified surface. The label will be retained by Zemax as surfaces are inserted or deleted around the target surface. The syntax is

```
SetLabel, surf, label
```
The returned value is the label. See also GetLabel and FindLabel.

SetMulticon
This item sets data in the multi-configuration editor. To set multi-configuration data, the syntax is
```
SetMulticon, config, row, value, status, pickuprow, pickupconfig, scale,
offset
```
If the config number is zero, SetMulticon may be used to set the operand type and number data using this syntax:
```
SetMulticon, 0, row, operand_type, number1, number2, number3
```
The returned string is formatted as defined for "GetMulticon". The numbers are used as described in "Summary of multi-configuration operands".
See also GetMulticon.

SetNSC ObjectData
This item sets the various data for NSC objects. The syntax is:
```
SetNSCObjectData, surf, object, code, data
```
Surf and object refer to the surface number and object number. The code is one of the integer values defined in the table in the discussion for "GetNSCObjectData". The returned value is identical to that returned by GetNSCObjectData after the new data is accepted. See also SetNSCObjectFaceData.

SetNSCObjectFaceData
This item sets the various data for NSC object faces. The syntax is:
```
SetNSCObjectFaceData, surf, object, face, code, data
```
Surf and object refer to the surface number and object number. Face is in the integer face number. The code is one of the integer values defined in the table in the discussion for "GetNSCObjectFaceData". The returned value is identical to that returned by GetNSCObjectFaceData after the new data is accepted.

SetNSCPosition
This item sets the position data for NSC objects. The syntax is:
```
SetNSCPosition, surf, object, code, data
```
Surf and object refer to the surface number and object number. Code is 1-7 for x, y, z, tilt-x, tilt-y, tilt-z, and material, respectively. Data is either a number (for codes 1-6) or a string with the material name (for code 7). The returned value is that of GetNSCPosition after the new data has been applied. See also GetNSCPosition.

SetNSCParameter
This item sets the parameter data for NSC objects. The syntax is:
```
SetNSCParameter, surf, object, parameter, data
```
Surf and object refer to the surface number and object number. Parameter is the integer parameter number, and data is the new numeric value for that parameter. The returned value is that of GetNSCParameter after the new data has been applied. See also GetNSCParameter.

SetNSCProperty
This item mimics the ZPL keyword SETNSCPROPERTY (see "SETNSCPROPERTY"). The syntax is:
```
SetNSCProperty, surface, object, code, face, value
```
Surface and object refer to the surface number and object number, and code refers to the integer code used to identify specific types of NSC object date as described in "SETNSCPROPERTY". The face argument is used by some codes to identify a particular object face. The value is either a string or a numeric value, depending upon the property being set. The return value is the same as for GetNSCProperty and corresponds to the newly set data.
See also GetNSCProperty

SetNSCSettings
This item sets the maximum number of intersections, segments, nesting level, minimum absolute intensity, minimum relative intensity, glue distance, miss ray distance, and ignore errors flag used for NSC ray tracing. The syntax is:

```
SetNSCSettings, MaxInt, MaxSeg, MaxNest, MinAbsI, MinRelI, GlueDist,
MissRayDist, IgnoreErrors
```
The returned value is identical to that of GetNSCSettings. Since the MaxSeg value may require large amounts of RAM, verify that the new value was accepted by checking the returned string value.
See also GetNSCSettings.

SetNSCSolve
This item sets the solve type on NSC position and parameter data. The syntax is:
```
SetNSCSolve, surface, object, parameter, solvetype, pickupobject,
pickupcolumn, scale, offset
```
For the x, y, and z position and tilt x, y, and z data, use a parameter value of -1 through -6, respectively. The solvetype is 0 for fixed, 1 for variable, 2 for pickup. If the solvetype is a pickup, the remaining arguments are the pickup object number, pickup column number (leave as 0 for the current column), and the scale and offset values. The returned value is identical to that of GetNSCSolve.
See also GetNSCSolve.

SetOperand
This item sets the operand data in the Merit Function Editor. The syntax is:
```
SetOperand, row, column, value
```
Row is the operand row number in the Merit Function Editor. Column is 1 for type, 2 for int1, 3 for int2, 4-7 for data1-data4, 8 for target, 9 for weight, and 10-11 for data 5-6. The returned string is the same as for GetOperand. To set the operand type, use column = 1, and for the value use the name of the operand, such as "EFFL".
To update the merit function after calling SetOperand, use the Optimize item with the number of cycles set to -1. See also GetOperand and Optimize.

SetPolState
This item sets the default polarization state set by the user. The data is formatted as follows:
```
SetPolState, nIsPolarized, Ex, Ey, Phax, Phay
```
See GetPolState for a complete description.

SetSettingsData
This item sets the settings data used by a window in temporary storage before calling MakeGraphicWindow or MakeTextWindow. The data may be retrieved using GetSettingsData. The syntax is:
```
SetSettingsData, number, data
```
Currently, only number = 0 is supported. This number may be used to expand the feature in the future. See also "How Zemax calls the client" and GetSettingsData.

SetSolve
This item sets solve data and returns data about the new solve on any surface. The syntax is:
```
SetSolve,surface,code,data...
```
where the code is as defined in the discussion of "GetSolve" and the "data..." are the exact parameters as defined for GetSolve. The parameter list depends upon the code listed. The return string is the same as GetSolve returns for the newly defined solve data.

SetSurfaceData
This item sets surface data. The item name is formatted as follows:
```
SetSurfaceData, surf, code, data
```
or
```
SetSurfaceData, surf, code, data, arg2
```
Where the values for code are the same as documented for the GetSurfaceData item. To set the surface type to a user defined surface, send the new DLL name using code 9 rather by setting the surface type. The return string is the newly set surface data formatted as documented for the GetSurfaceData item. See also GetSurface- Data and SetSurfaceParameter.

SetSurfaceParameter

This item sets surface parameter data. The item name is formatted as follows:

```
SetSurfaceParameter, surf, parm, newvalue
```

where surf is the surface number and parm is the parameter number. The return value is identical to GetSurfaceParameter. See also SetSurfaceData and GetSurfaceParameter.

SetSystem

This item sets system data. The syntax is

```
SetSystem, unitcode, stopsurf, rayaimingtype, useenvdata, temp,
pressure, globalrefsurf
```

The returned string is formatted as defined for GetSystem. See also GetSystem for details on the arguments. The value for useenvdata is ignored.

SetSystemAper

This item sets system aperture data. The syntax is

```
SetSystemAper, type, stopsurf, aperture_value
```

The returned string is formatted as defined for GetSystemAper. See GetSystemAper for details on the arguments.

SetSystemProperty

This item mimics the ZPL keyword SETSYSTEMPROPERTY. The syntax is:

```
SetSystemProperty code, value1, value2
```

Code is an integer value that defines the specific system property requested. For a listing of these codes, see "SETSYSTEMPROPERTY, SYSP". The return string format is the same as for GetSystemProperty.
See also GetSystemProperty.

SetTol

This item sets tolerance operand data. The syntax is

```
SetTol, n, col, value
```

where n is the tolerance operand number, col is 1 for type, 2-4 for int1-int3, 5 for min, and 6 for max. The returned string is formatted as for GetTol.
See also GetTol.

SetUDOData

Obsolete, see SetUDOItem below.

SetUDOItem

This item is used to pass just one datum computed by the client program to the Zemax optimizer. The only time this item name should be used is when implementing a User Defined Operand, or UDO. UDO's are described in "Optimizing with externally compiled programs". The syntax for SetUDOItem is

```
SetUDOItem, buffercode, datanumber, data
```

where buffercode is an integer passed to the UDO on the command line and data is the data item number being passed. After the last data item has been sent, the buffer must be closed using the CloseUDOData item before the optimization may proceed. A typical implementation may consist of the following series of item messages:

```
SetUDOItem, buffercode, 0, value0
SetUDOItem, buffercode, 1, value1
SetUDOItem, buffercode, 2, value2
CloseUDOData, buffercode
```

See also GetUDOSystem.

SetVig

This item sets the vignetting factors. For more information, see "Vignetting factors". No arguments are required. SetVig always returns "OK" as the return string.

SetWave

This item sets wavelength data. The syntax is
```
 SetWave, 0, primary, number
```
or
```
 SetWave, n, wavelength, weight
```
If the value for n is zero, then the primary wavelength number and the total number of wavelengths is set to the new integer values. If n is a valid wavelength number (between 1 and the number of wavelengths, inclusive) then the wavelength in micrometers and the wavelength weight are both set. See also GetWave. The returned data is the same as for GetWave, n.

WindowMaximize

This item will maximize the main Zemax window or any analysis window Zemax currently displays. The syntax is
```
 WindowMaximize, winnum
```
Use 0 for winnum to maximize the main Zemax window, otherwise, the integer window number.

WindowMinimize

This item will minimize the main Zemax window to an icon. See WindowMaximize for details.

WindowRestore

This item will restore the main Zemax window to it's previous size and position. See WindowMaximize for details.

Tracing large numbers of rays

If only a few rays need to be traced, then it is easy and fast enough to use DDE items such as GetTrace or GetPolTrace. However, if more than perhaps 100-500 rays need to be traced, it is far faster to use the array ray tracing technique.

Instead of tracing the rays one at a time, an array containing a list of all the rays to be traced is created, and then the entire array passed to Zemax at one time. Zemax then traces all the rays, and passes the entire array back to the client.

This method mimics the behavior of GetTrace, GetTraceDirect, GetPolTrace, and GetPolTraceDirect, with the added power that any number of rays to be traced at one time, although it is slightly more complex to program.

There are 2 steps required:

1) Fill an array of structures with the required data
2) Pass the array to Zemax

Step 1: Placing the ray data in the array

The data defining the list of rays to be traced is placed in an array of structures of the following form:
```
 typedef struct
       {
       double x, y, z, l, m, n, opd, intensity;
       double Exr, Exi, Eyr, Eyi, Ezr, Ezi;
       int wave, error, vigcode, want_opd;
       }DDERAYDATA;
```

Inside the client program, an array of type DDERAYDATA is created with a declaration such as:
```
 DDERAYDATA MyData[101];
```

There are four trace modes supported, numbered from 0 to 3, and they correspond to the actions of GetTrace (mode 0), GetTraceDirect (mode 1), GetPolTrace (mode 2), and GetPolTraceDirect (mode 3). Each mode requires a slightly different set of data to be placed in the DDERAYDATA structure. The mode is set by the opd parameter in array position zero.

Each ray requires 1 array element. If there are "n" rays to be traced, then the array structures 1 through n will be used to hold the data for these n rays. Array position zero is reserved for some header data that needs to be passed with the array.

Mode 0: Similar to GetTrace

Like GetTrace, the rays are defined by the relative field and pupil coordinates, the wavelength, the mode (either real, mode = 0 or paraxial, mode = 1) as well as the surface to trace the ray to. Before the rays can be traced, the array should be modified in array positions 1 through n to contain the definitions for each ray. The field, pupil, and wavelength data are all stored "in place" in the array as follows:

```
MyData[i].x = hx;
MyData[i].y = hy;
MyData[i].z = px;
MyData[i].l = py;
MyData[i].m = 0.0; /* ignored on input */
MyData[i].n = 0.0; /* ignored on input */
MyData[i].opd = 0.0; /* ignored on input */
MyData[i].intensity = 1.0; /* initial intensity */
MyData[i].Exr = 0.0; /* on output, x cosine of the surface normal */
MyData[i].Exi = 0.0; /* ignored */
MyData[i].Eyr = 0.0; /* on output, y cosine of the surface normal */
MyData[i].Eyi = 0.0; /* ignored */
MyData[i].Ezr = 0.0; /* on output, z cosine of the surface normal */
MyData[i].Ezi = 0.0; /* ignored */
MyData[i].wave = wavenumber;
MyData[i].error = 0; /* must initially be zero */
MyData[i].vigcode = 0; /* must initially be zero */
MyData[i].want_opd = 0; /* 0 if OPD data is not needed, 1 if it is */
```

Note the argument i should be between 1 and n, the number of rays. The array may be of any size, however, it is bad programming practice to trace too many rays at once, because it makes it difficult to accept a user interrupt if the ray tracing is to be terminated. The initial intensity will be scaled by any pupil apodization or surface apodization that occurs. Note mode 0 considers pupil apodization, while mode 1 does not.

The array position 0 is reserved for other data that applies to all the rays. The data in array position 0 is used to tell Zemax the mode, the number of rays, and the surface to trace the rays to. The data is placed in the array as follows:

```
MyData[0].x = 0.0; /* ignored */
MyData[0].y = 0.0; /* ignored */
MyData[0].z = 0.0; /* ignored */
MyData[0].l = 0.0; /* ignored */
MyData[0].m = 0.0; /* ignored */
MyData[0].n = 0.0; /* ignored */
MyData[0].opd = 0; /* sets mode 0 */
MyData[0].intensity = 0.0; /* ignored */
MyData[0].Exr = 0.0; /* ignored */
MyData[0].Exi = 0.0; /* ignored */
MyData[0].Eyr = 0.0; /* ignored */
MyData[0].Eyi = 0.0; /* ignored */
MyData[0].Ezr = 0.0; /* ignored */
MyData[0].Ezi = 0.0; /* ignored */
MyData[0].wave = mode; /* 0 for real rays, 1 for paraxial rays */
MyData[0].error = numrays; /* the number of rays in the array */
MyData[0].vigcode = 0; /* must initially be zero */
MyData[0].want_opd = lastsurf; /* -1 for image, or any valid surface number */
```

The number of rays must be placed in the error variable. Note that any declaration for an array of DDERAY- DATA types must be 1 bigger than the number of rays, since array position 0 is reserved for the data described above. All rays in the array must be traced to the same surface using the same mode, but they all may have different wavelengths and field and pupil coordinates.

If the variable want_opd is 0, then the usual x, y, z, l, m, and n data will be computed for the ray. If want_opd is anything other than zero, then the OPD for the ray will be returned along with the usual data. Computing the OPD takes additional time beyond that for regular ray tracing, and Zemax only performs this additional computation if requested to do so. Computing the OPD is also more complicated for the client program, because OPD means optical path difference; which means two rays must be traced rather than just one. To compute OPD for an arbitrary ray, Zemax must trace the chief ray, then the arbitrary ray, then subtract the phase of the two to get the OPD. Rather than trace the same chief ray over and over, which is slow, usually the chief ray is traced once, and then the phase of the chief ray is subtracted from each

subsequent ray. Zemax uses the sign of the want_opd parameter to specify which calculation should be done. If want_opd is less than zero (such as -1) then the both the chief ray and specified ray are requested, and the OPD is the phase difference between the two in waves of the current wavelength. If want_opd is greater than zero, then the most recently traced chief ray data is used. Therefore, the want_opd flag should be -1 whenever the chief ray changes; and +1 for all subsequent rays which do not require the chief ray be traced again. Generally the chief ray changes only when the field coordinates or wavelength changes. This method is much faster if there are many rays being traced from the same field point, as is the case for many optical analysis calculations.

The opd can only be computed if the last surface is the image surface, otherwise, the opd value will be zero.

Mode 1: Similar to GetTraceDirect

Like GetTraceDirect, the rays are defined by x, y, z, l, m, and n coordinates on any starting surface, as well as the wavelength, the mode (either real, mode = 0 or paraxial, mode = 1) and the surface to trace the ray to. Before the rays can be traced, the array should be modified in array positions 1 through n to contain the definitions for each ray. The field, pupil, and wavelength data are all stored "in place" in the array as follows:

```
MyData[i].x = x;
MyData[i].y = y;
MyData[i].z = z;
MyData[i].l = l;
MyData[i].m = m;
MyData[i].n = n;
MyData[i].opd = 0.0; /* ignored on input */
MyData[i].intensity = 1.0; /* initial intensity */
MyData[i].Exr = 0.0; /* on output, x cosine of the surface normal */
MyData[i].Exi = 0.0; /* ignored */
MyData[i].Eyr = 0.0; /* on output, y cosine of the surface normal */
MyData[i].Eyi = 0.0; /* ignored */
MyData[i].Ezr = 0.0; /* on output, z cosine of the surface normal */
MyData[i].Ezi = 0.0; /* ignored */
MyData[i].wave = wavenumber;
MyData[i].error = 0.0; /* must initially be zero */
MyData[i].vigcode = 0.0; /* must initially be zero */
MyData[i].want_opd = 0; /* ignored */
```

Note the argument i should be between 1 and n, the number of rays. The array may be of any size, however, it is bad programming practice to trace too many rays at once, because it makes it difficult to accept a user interrupt if the ray tracing is to be terminated. The initial intensity will be scaled by any surface apodization that occurs. Note mode 0 considers pupil apodization, while mode 1 does not.

The array position 0 is reserved for other data that applies to all the rays. The data in array position 0 is used to tell Zemax the mode, the number of rays, and the surface to trace the rays to. The data is placed in the array as follows:

```
MyData[0].x = 0.0; /* ignored */
MyData[0].y = 0.0; /* ignored */
MyData[0].z = 0.0; /* ignored */
MyData[0].l = 0.0; /* ignored */
MyData[0].m = 0.0; /* ignored */
MyData[0].n = 0.0; /* ignored */
MyData[0].opd = 1; /* sets mode 1 */
MyData[0].intensity = 0.0; /* ignored */
MyData[0].Exr = 0.0; /* ignored */
MyData[0].Exi = 0.0; /* ignored */
MyData[0].Eyr = 0.0; /* ignored */
MyData[0].Eyi = 0.0; /* ignored */
MyData[0].Ezr = 0.0; /* ignored */
MyData[0].Ezi = 0.0; /* ignored */
MyData[0].wave = mode; /* 0 for real rays, 1 for paraxial rays */
MyData[0].error = numrays; /* the number of rays in the array */
MyData[0].vigcode = startsurf; /* the surface on which the coordinates start */
MyData[0].want_opd = lastsurf; /* -1 for image, or any valid surface number */
```

The behavior of this mode is otherwise the same as in mode 0. Computation of the OPD is not permitted in this mode.

Mode 2: Similar to GetPolTrace

This mode is very similar to GetPolTrace. Before the rays can be traced, the array should be modified in array positions 1 through n to contain the definitions for each ray. The field, pupil, and wavelength data are all stored "in place" in the array as follows:

```
MyData[i].x = hx;
MyData[i].y = hy;
MyData[i].z = px;
MyData[i].l = py;
MyData[i].m = 0.0; /* ignored */
MyData[i].n = 0.0; /* ignored */
MyData[i].opd = 0.0; /* ignored */
MyData[i].intensity = 1.0; /* initial intensity */
MyData[i].Exr = Exr /* Electric field X real */
MyData[i].Exi = Exi /* Electric field X imaginary */
MyData[i].Eyr = Eyr /* Electric field Y real */
MyData[i].Eyi = Eyi /* Electric field Y imaginary */
MyData[i].Ezr = Ezr /* Electric field Z real */
MyData[i].Ezi = Ezi /* Electric field Z imaginary */
MyData[i].wave = wavenumber;
MyData[i].error = 0.0; /* must initially be zero */
MyData[i].vigcode = 0.0; /* must initially be zero */
MyData[i].want_opd = 0; /* ignored */
```

Note the argument i should be between 1 and n, the number of rays. The array may be of any size, however, it is bad programming practice to trace too many rays at once, because it makes it difficult to accept a user interrupt if the ray tracing is to be terminated.

The intensity parameter is used to return the relative transmitted intensity of the ray. If all six of the electric field values are zero; then Zemax will use the Ex and Ey values provided in array position 0 to determine the electric field. Otherwise, the electric field is defined by these six values. The defined electric field vector must be orthogonal to the ray vector or incorrect ray tracing will result. Even if these six values are defined for each ray, values for Ex and Ey in the array position 0 must still be defined, otherwise an unpolarized ray trace will result.

The array position 0 is reserved for other data that applies to all the rays. The data in array position 0 is used to tell Zemax the mode, the number of rays, and the surface to trace the rays to. The data is placed in the array as follows:

```
MyData[0].x = Ex; /* Electric field amplitude in x */
MyData[0].y = Ey; /* Electric field amplitude in y */
MyData[0].z = Phax; /* Phase in degrees for Ex */
MyData[0].l = Phay; /* Phase in degrees for Ey */
MyData[0].m = 0.0; /* ignored */
MyData[0].n = 0.0; /* ignored */
MyData[0].opd = 2; /* sets mode 2 */
MyData[0].intensity = 0.0; /* ignored */
MyData[0].Exr = 0.0; /* ignored */
MyData[0].Exi = 0.0; /* ignored */
MyData[0].Eyr = 0.0; /* ignored */
MyData[0].Eyi = 0.0; /* ignored */
MyData[0].Ezr = 0.0; /* ignored */
MyData[0].Ezi = 0.0; /* ignored */
MyData[0].wave = mode; /* 0 for real rays, 1 for paraxial rays */
MyData[0].error = numrays; /* the number of rays in the array */
MyData[0].vigcode = 0; /* must initially be zero */
MyData[0].want_opd = lastsurf; /* -1 for image, or any valid surface number */
```

The behavior of this mode is otherwise the same as in mode 0.

Mode 3: Similar to GetPolTraceDirect

Like GetPolTraceDirect, the rays are defined by x, y, z, l, m, and n coordinates on any starting surface, as well as the wavelength, the mode (either real, mode = 0 or paraxial, mode = 1) and the surface to trace the ray to. Before the rays can be traced, the array should be modified in array positions 1 through n to contain the definitions for each ray. The field, pupil, and wavelength data are all stored "in place" in the array as follows:

```
MyData[i].x = x;
MyData[i].y = y;
MyData[i].z = z;
MyData[i].l = l;
MyData[i].m = m;
MyData[i].n = n;
MyData[i].opd = 0.0; /* ignored */
MyData[i].intensity = 1.0; /* initial intensity */
MyData[i].Exr = Exr /* Electric field X real */
MyData[i].Exi = Exi /* Electric field X imaginary */
MyData[i].Eyr = Eyr /* Electric field Y real */
MyData[i].Eyi = Eyi /* Electric field Y imaginary */
MyData[i].Ezr = Ezr /* Electric field Z real */
MyData[i].Ezi = Ezi /* Electric field Z imaginary */
MyData[i].wave = wavenumber;
MyData[i].error = 0.0; /* must initially be zero */
MyData[i].vigcode = 0.0; /* must initially be zero */
MyData[i].want_opd = 0; /* ignored */
```

Note the argument i should be between 1 and n, the number of rays. The array may be of any size, however, it is bad programming practice to trace too many rays at once, because it makes it difficult to accept a user interrupt if the ray tracing is to be terminated.

The intensity parameter is used to return the relative transmitted intensity of the ray. If all six of the electric field values are zero; then Zemax will use the Ex and Ey values provided in array position 0 to determine the electric field. Otherwise, the electric field is defined by these six values. The defined electric field vector must be orthogonal to the ray vector or incorrect ray tracing will result. Even if these six values are defined for each ray, values for Ex and Ey in the array position 0 must still be defined, otherwise an unpolarized ray trace will result.

The array position 0 is reserved for other data that applies to all the rays. The data in array position 0 is used to tell Zemax the mode, the number of rays, and the surface to trace the rays to. The data is placed in the array as follows:

```
MyData[0].x = Ex; /* Electric field amplitude in x */
MyData[0].y = Ey; /* Electric field amplitude in y */
MyData[0].z = Phax; /* Phase in degrees for Ex */
MyData[0].l = Phay; /* Phase in degrees for Ey */
MyData[0].m = 0.0; /* ignored */
MyData[0].n = 0.0; /* ignored */
MyData[0].opd = 3; /* sets mode 3 */
MyData[0].intensity = 0.0; /* ignored */
MyData[i].Exr = 0.0; /* ignored */
MyData[i].Exi = 0.0; /* ignored */
MyData[i].Eyr = 0.0; /* ignored */
MyData[i].Eyi = 0.0; /* ignored */
MyData[i].Ezr = 0.0; /* ignored */
MyData[i].Ezi = 0.0; /* ignored */
MyData[0].wave = mode; /* 0 for real rays, 1 for paraxial rays */
MyData[0].error = numrays; /* the number of rays in the array */
MyData[0].vigcode = startsurf; /* the surface on which the coordinates start */
MyData[0].want_opd = lastsurf; /* -1 for image, or any valid surface number */
```

The behavior of this mode is otherwise the same as in mode 1.

Mode 5: For tracing non-sequential rays

Mode 5 is used to define and trace a ray inside of a non-sequential group. Unlike the other modes; there is no text based command for tracing NSC rays, and only 1 ray at a time may be traced. However, rays may split or scatter into multiple paths, and mode 5 will trace them all and return the entire tree of ray data.

To define the starting ray, use array position zero. The data is placed in the array as follows:

```
MyData[0].x = 0.0; /* starting x coordinate */
MyData[0].y = 0.0; /* starting y coordinate */
MyData[0].z = 0.0; /* starting z coordinate */
MyData[0].l = 0.0; /* starting x direction cosine */
MyData[0].m = 0.0; /* starting y direction cosine */
```

```
MyData[0].n = 1.0; /* starting z direction cosine */
MyData[0].opd = 5+nMaxSegments; /* sets mode 5, see comments below */
MyData[0].intensity = 1.0; /* initial intensity */
MyData[0].Exr = 0.0; /* initial E field if doing pol ray tracing */
MyData[0].Exi = 0.0;
MyData[0].Eyr = 0.0;
MyData[0].Eyi = 0.0;
MyData[0].Ezr = 0.0;
MyData[0].Ezi = 0.0;
MyData[0].wave = 0; /* wavelength number, use 0 for randomly selected by weight */
MyData[0].error = 1; /* NSC group surface, 1 if the program mode is NSC */
MyData[0].vigcode = 0; /* controls polarization, split, and scatter */
MyData[0].want_opd = 0; /* the inside of flag, use 0 if ray is not inside anything */
```

Only array position zero is used to define the starting ray. Zemax will trace the ray through the NSC group and fill the remaining array positions with ray data. Since rays may split or scatter into multiple child rays, a very large number of segments may be returned. The nMaxSegments number is the maximum allowed size of the array for Zemax to return. The array should also be large enough to hold at least nMaxSegments elements. The nMaxSegments value Zemax is using for the current optical system may be determined by the GetNSCSettings command.

If polarization ray tracing is used, the initial electric field values must be provided. The user application must ensure the defined vector is orthogonal to the ray propagation vector, and that the resulting intensity matches the starting intensity value, otherwise incorrect ray tracing results will be produced.

The integer vigcode value determines if polarization ray tracing, splitting, and scattering are to be used. To determine the vigcode value, use 0 for no polarization, 1 for polarization, plus 2 if splitting is used, plus 4 if scattering is used. The resulting integer will be between 0 and 7 inclusive. Note if ray splitting is to be used, polarization must be used as well.

The want_opd flag indicates where the ray starts. If zero, then the ray is assumed to not start in any object. Otherwise, the ray starts inside the media defined by the specified object number. This is identical to the "inside of" flag set on the NSC Editor in Zemax.

The array is passed to Zemax just like the other modes, see "Step 2: Pass the array to Zemax". When the ray tracing is complete, the array will contain the entire ray tree. Array element zero also contains the total number of segments stored in the array element want_opd:

```
nNumRaySegments = MyData[0].want_opd;
```

The other values are interpreted as follows:
```
i = the segment number
MyData[i].wave = segment level
MyData[i].want_opd = segment parent
MyData[i].vigcode = inside of object number
MyData[i].error = hit object number
MyData[i].x,.y,.z,.l,.m,.n = the ray coordinates and cosines
MyData[i].intensity = the ray intensity
MyData[i].opd = the path length to the hit object
```

See also GetNSCSettings and SetNSCSettings.

Step 2: Pass the array to Zemax
Once all the rays are defined, the array must be sent to the Zemax server. The easy way to do this is using the ZCLIENT PostArrayTraceMessage function:

```
PostArrayTraceMessage(szBuffer, RD);
```

The RD variable is the pointer to the DDERAYDATA array holding the list of rays. The Zemax server will trace all the rays, then pass the data back in the same array. For more details on the code sample, see the section "a sample client program" at the end of the chapter.

Note that an extension should never request that more rays be traced at one time than can be traced within the timeout period specified in the zclient.c code. If more rays are traced in one call to PostArrayTraceMessage than can be traced in the time allowed by DDE_TIMEOUT, an error will occur, and PostArrayTraceMessage will return -1. If the ray trace is successful, PostArrayTraceMessage will return 0. It is good programming practice to trace rays in small groups of 1000-

5000 rays at a time. Once this many rays are traced, the overhead of multiple calls using PostArrayTraceMessage becomes negligible.

For information about timeout limits when executing potentially long calculations, see "Comments on the DDE timeout interval".

How Zemax calls the client

Zemax looks in the <program>\Extend folder (see "Folders") for Extension executables. Each program in this folder ending in the usual .EXE extension is assumed to be a valid Zemax Extension program. The name of the executable is placed in the pull down menu under "Extension List" in the Extensions section of the Programming tab.

When any of the listed Extension names is selected from the menu, Zemax executes the Extension program with the following syntax:

programname textflag optionsflag tempfile {settings data}

The programname is the full path to the executable, for example, "C:\Zemax\EXTENSIONS\DDE_DE- MO.EXE".

The textflag is initially always 0. The convention Zemax uses for Extensions is the same as for internal Zemax features: if the feature generates both a graphic and a text format for output, show the graphic version first. If the feature does not have a graphic display, then show a text window. The Extension must generate a graphic file if it can and is requested to do so. The user can click on "text" to look at the text version in a new window; if this happens Zemax will call the Extension again with the text flag set to 1. If the Extension only generates text, then it can generate a text file and open it by sending the MakeTextWindow item. Zemax will never send the textflag set to 1 unless the Extension specifically states it can support a text mode; this is why the MakeGraphicWindow returns a flag to indicate if the feature supports text, and why Zemax always ask for a graphic mode first.

The optionsflag is set to 0 if the user does not wish to see the "settings" box; or 1 if the user does. There are two ways this flag may be set; either the user clicked on "Settings" on a window already open, or, the user has selected "Show Options First" from the Environment options within Zemax. In this latter case the settings box may need to be shown before the calculation is done the first time. If this flag is set the Extension must prompt the user for setting options using a dialog box or other interface mechanism. If there are no user defined options, then a message box stating "This window has no options" should be displayed.

The tempfile is the fully qualified path and file name of the temporary file to hold the data generated by the Extension. Initially, this file will not exist, and the Extension needs to create it, write either text or graphic data to the file, and then call MakeTextWindow or MakeGraphicWindow as appropriate. If the user is updating an existing window, then the file may already exist, and it should be overwritten. The file should never be opened and read as the data may not be what the Extension originally put there. Zemax does some translation of the Extension created graphic file in particular. No other file should be used for output, because Zemax expects windows which are being updated to retain their original temporary file names. Only one window may be created by a single call to the Extension.

When Zemax receives a MakeTextWindow or MakeGraphicsWindow request from the Extension, one of the arguments encoded in the atom name is a space delimited string of Extension defined values. This string should be used to hold the "settings data" for the settings the Extension requires, such as the surface number, field, wavelength, or other data for which the data in the window was computed. When Zemax updates a window, it will call the Extension and provide the settings data string back to the Extension.

It is important that the Extension does not attempt to store the settings data itself, either statically or in a file. The reason is that Zemax may have multiple text and graphic windows open at once, all using the same Extension code to generate the data displayed. For example, a Extension that computes and displays the power of a single surface may be used by several windows simultaneously, each for a different surface. Only Zemax "knows" which window is using which setting data; and so Zemax passes this data on the command line. On the initial call to the Extension, the settings data string is not supplied, and the Extension must determine it's own default settings.

Because DDE data items cannot be longer than 255 characters, it is a good idea to use the SetSettingsData rather than return the settings data appended to the end of the MakeGraphicsWindow or MakeTextWindow commands. If the extension uses SetSettingsData to store the settings data with the window, then GetSettings- Data should be used to retrieve the data whenever needed.

Generating a text window

Generating a text window for display by Zemax is accomplished with the MakeTextWindow item. First, open the filename passed to the Extension using the typical C language fopen function, with "wt" as the mode for "write text".

Then, output data to the file, close the file, then send the MakeTextWindow item to Zemax. Zemax will read the data and display it without modification or interpretation. A test code might look like:

```
FILE *out;
out = fopen(szFile, "wt");
fputs("Write this line\n", out);
fclose(out);
```

Do not forget to close the file!

Generating a graphic window

The Extension generates a graphic window by writing out graphic format data to the temporary file name provided on the command line. The graphic data is created using a script language similar to that used by the ZPL macro language or the annotation feature.

All Zemax graphics are written on a virtual screen whose coordinates are 1.00 wide by 1.00 high, with the 0, 0 origin at the lower left corner. The screen aspect ratio is normally 4 wide by 3 high, or 5 wide by 3 high, depending upon user preference. Therefore, the virtual pixels being drawn are wider than they are tall.

Each line in the script file should be followed by a newline "\n" character. After writing all of the graphic commands to the file, the file should be closed for writing.

The following graphic script commands are supported.

ADDRESS

Syntax: ADDRESS

The address command will cause the "address box" in the lower right edge of the plot to appear, unless the user has selected this box not to appear.

BOX

Syntax: BOX x1 y1 x2 y2

The BOX command will draw four lines, connecting the four points of a rectangle. The coordinates are the lower left and upper right coordinates of the rectangle.

DATA

Syntax: DATA n "string"

The DATA command accepts an integer argument between 0 and 5, and a text string. The integer argument indicates on which row of the data box the string should be written. Normally, the lens name goes on row 0; and the date on row 1. The LENSNAME and DATE commands do this automatically. The DATA command allows any string to be placed on any of the lines.

DATE

Syntax: DATE

The current data and time is normally written on the second line of the data box below the plot title. This command has no arguments, and will print the date and/or time (set in the General section of the Project Preferences) on this second line.

FRAME

Syntax: FRAME

Either a FRAME or a NOFRAME command must be the first line in the script file. FRAME will cause a standard Zemax graphics frame to be drawn, with a rectangle around the entire perimeter, and two ruling lines at about 0.2 in the y direction. The plot title can be written centered inside these ruling lines using the TITLE command. See also NOFRAME.

GRID

Syntax: GRID x1 y1 x2 y2 nx ny

The GRID command creates a grid of nx by ny grey boxes. This command is typically used to make the background grid lines that mimics the style of most Zemax graphics. The coordinates are the lower left and upper right coordinates of the rectangle. See also BOX.

LENSNAME

Syntax: LENSNAME
The current lens name is normally written on the first line of the data box below the plot title. This command has no arguments, and will print the lens name on this line.

LINE

Syntax: LINE x1 y1 x2 y2
This is the most basic graphic command; it draws a line connecting the two points. Note that the coordinates should be between 0 and 1.0.

NOFRAME

Syntax: NOFRAME
Either a FRAME or a NOFRAME command must be the first line in the script file. NOFRAME will cause a standard Zemax graphics frame to be drawn, but without the two ruling lines. See also FRAME.

PENSTYLE

Syntax: PENSTYLE ncolor nstyle nwidth
The PENSTYLE command changes the current pen color, style, and width. The integer ncolor must be 0 for black, or between 1 and 12, for the corresponding pen color. The colors may be defined by the user in the Colors section of the Project Preferences (in the Setup Tab). The nstyle integer can be 0 for a solid line, or an integer between 1 and 4 for various styles of dashed lines. The integer nwidth is the relative width of the pen. The default value is 1. A value of 2 will make lines twice as wide as the default width. If the pen style is not a solid line (nstyle = 0) then the pen width must be 1 pixel; Windows does not support thick lines which are not solid. All subsequent text and lines drawn will be in the new color, style, and width.

TEXT

Syntax: TEXT "string" x y angle width height
The text command is used to place text data on the graphic. The x and y coordinates are the starting coordinates for the string. The angle is in degrees. The width and height values affect the font size. The default value of 10 for each yields the same font size Zemax normally uses; which is 1/70 of the screen width and 1/40 of the screen height. Numbers larger or smaller than 10 will yield proportionally scaled font sizes.

TITLE

Syntax: TITLE "string"
If FRAME was used to define the graphic rather than NOFRAME, then the plot title would normally be centered within the ruling lines along the bottom 20% of the plot. The "string" provided will be displayed in this position by the TITLE command.

A sample Extension program

Within the <program>\Extend folder (see "Folders") is a sample application called DDE_DEMO which illustrates the use of DDE with Zemax. The source code is included. DDE_DEMO can operate in either of two modes: stand alone, or as an Extension.
If run directly from a command prompt, Windows Explorer, or by double-clicking on the DDE_DEMO icon, it will run as a stand alone application, conducting the DDE in the background. The data it generates will be displayed in the DDE_DEMO window. To run DDE_DEMO in stand alone mode, first run Zemax, load a lens file, then launch DDE_DEMO by double clicking on the DDE_DEMO icon. Try loading different lens files, or editing the one in the LDE, and watch the DDE_DEMO window to monitor the changes. DDE_DEMO will list out some of the system, surface, field, and wavelength data, and will trace a few rays, and list the x, y, and z coordinates of the rays on the screen. The program automatically updates itself every few seconds.

To run DDE_DEMO as an Extension, run Zemax, load a lens file, then select Extension List (from the Programming tab), then DDE_DEMO. Note that any .EXE application placed in the <program>\Extend folder (see "Folders") will be listed in the Zemax Extension List menu; whether or not it is a valid stand alone or Extension capable of communicating with Zemax. DDE_DEMO will execute in a "hidden" window, establish the DDE link with Zemax, extract the system and ray data, write the data to a file, notify Zemax that the file is ready (using the MakeTextWindow item) and then close. The client's job is finished once it has generated the data; and it should terminate itself when done.

When Zemax receives the MakeTextWindow item, it reads the contents of the file and displays the data in a standard Zemax text window. If the window is now "updated", or the "settings" option is selected, Zemax will call the client program again and ask for the data to be computed again.

Note that the exact same program can act as both a stand alone client and as an Extension. The secret to the bimodal operation is that if Zemax calls the client, it passes command line arguments to the client telling it what type of data to create and where to write the data to. If no command line arguments are provided, the client can assume it was launched from the operating system, and therefore it should run in stand alone mode. The DDE exchange and subsequent processing is the same either way; only a few key lines of code are required to determine whether the window should be visible or hidden, and whether the data is written to a file or the client's own screen.

The DDE_DEMO code

The source code for DDE_DEMO is provided. It contains the basic algorithms, variables, declarations, and Windows message handling required of all client programs. There are portions of the code which illustrate the use of WM_DDE_POKE for tracing arrays of rays as well.

DDE_DEMO is a rather difficult program to use as a starting point for an Extension, although it does show some high level techniques. A simpler way to implement Zemax Extensions is described below.

A simplified technique for implementing Extensions

Obviously, DDE programming for Windows is not for the faint of heart; it involves numerous programming skills and significant familiarity with message loops, DDE, pointers, atoms, and global handles.

However, most of the code used to communicate with Zemax is standard "boiler plate" code which is common to any DDE client program. This code has all been written already, and has been placed into a single source code module program called ZCLIENT. ZCLIENT is provided in source code form with Zemax, and may be freely copied (as long as the copyright is retained), and used in new Extensions.

The ZCLIENT program

ZCLIENT handles all of the communication with Zemax transparently. Embedded within ZCLIENT is a call to a single function given the name "UserFunction". This function is placed in a separate C program provided by the user, and is compiled along with ZCLIENT to create the Extension executable.

When Zemax calls the Extension, execution begins within ZCLIENT. ZCLIENT establishes the DDE commu- nication, then calls UserFunction. Within UserFunction, two functions provided by ZCLIENT, PostRequestMes- sage and GetString, are all that are usually required to get the data needed from Zemax. The data is then formatted as either a text or a graphic, and sent back to Zemax for display.

The syntax for PostRequestMessage is as follows:

```
PostRequestMessage(szItemName, szBuffer);
```

szItemName contains the name of the item desired; the available item names are described earlier in this chapter. The desired data is then sent back in the string szBuffer. This function returns 0 if the data transfer was successful, otherwise, it returns -1. Zemax normally provides comma delimited data, and the function GetString extracts the individual items.
The syntax for GetString is:

```
GetString(szBuffer, nItem, szSubString);
```

The szBuffer string is the buffer returned by PostRequestMessage. The integer nItem is the number of the item desired; 0 for the first item, 1 for the second, etc. The character szSubString holds the string from the nItem position.
For example, to get the name of the lens, the code would be:

```
PostRequestMessage("GetName", szBuffer);
GetString(szBuffer, 0, szLensName);
```

The string szLensName now contains the name of the lens. Because "GetName" only returns 1 item, this code could be shortened to just one line:

```
PostRequestMessage("GetName", szLensName);
```

There is also a modified version of PostRequestMessage for doing fast tracing of large numbers of rays. The function is PostArrayTraceMessage. This function accepts two arguments: the buffer string, and the address of the array of ray data. A sample array ray trace code using PostArrayTraceMessage is provided, it is called ARR_DEMO.C.
To trace rays using PostArrayTraceMessage, the ray list must first be defined, then a single call to PostArrayTraceMessage must be made. Here is a sample of the technique:

```
/* Fill RD array position 0 with the header data. */
RD[0].x = 0.0;
RD[0].y = 0.0;
RD[0].z = 0.0;
RD[0].l = 0.0;
RD[0].m = 0.0;
RD[0].n = 0.0;
RD[0].opd = 0.0; /* this is where we set the mode, mode 0 is like GetTrace */
RD[0].intensity = 0.0;
RD[0].wave = 0;
RD[0].error = 25; /* trace 25 rays */
RD[0].vigcode = 0;
RD[0].want_opd = -1;

/* Define the 25 rays. Obviously, you can define any rays you want... */
k = 0;
for (i = -2; i <= 2; i++)
      {
      for (j = -2; j <= 2; j++)
            {
            k++;
            RD[k].x = 0.0;
            RD[k].y = 0.0;
            RD[k].z = (double) i / 4.0;
            RD[k].l = (double) j / 4.0;
            RD[k].m = 0.0;
            RD[k].n = 0.0;
            RD[k].opd = 0.0;
            RD[k].intensity = 0.0;
            RD[k].wave = 1;
            RD[k].error = 0;
            RD[k].vigcode = 0;
            RD[k].want_opd = 0;
            }
      }
/* Now go get the data */
PostArrayTraceMessage(szBuffer, RD);

/* The output data is now stored in RD, ready to use! */
/* Note just one line of code to trace all rays. */
```

Using ZCLIENT is vastly easier than writing all the DDE communication code yourself. This sample array trace is taken from ARR_DEMO, one of the sample DDE codes that use ZCLIENT.
For information about timeout limits when executing potentially long calculations, see "Comments on the DDE timeout interval".

A sample Extension program using ZCLIENT

The source code for a complete Extension which uses ZCLIENT is provided. The Extension is called PHASPLOT, and it uses ZCLIENT.C as well as PHASPLOT.C. PHASPLOT generates either a graph or a text listing of the phase of a binary 2 surface. The code illustrates how to make a graphic and a text window, how to get data out of Zemax, and how to format output for graphics and text display.

PHASPLOT also has a Zemax style "settings" box, the source code for which is included in PHASPLOT.C in the "SurDlgProc" function; which uses the PHASPLOT.RC file to define the dialog box appearance.

Writing Extensions from the PhasePlot template

The best way to learn how to write Extensions is to start with PhasePlot, and start editing! Make a copy of PhasePlot.C and PhasePlot.RC, and rename the copies to something else (MY_CODE.C, MY_CODE.RC). Then create a new project using any of the good commercially available C compilers (Borland and MicroSoft both make excellent tools) and load these 3 files into the project. Recompile and build the project into a test EXE to make sure everything is in place. Then, edit the code within UserFunction and the settings dialog function to suit your requirements.

Getting analysis data using GetTextFile

One often overlooked capability of Extensions is the ability to create text and graphic data files from analysis already supported by Zemax by using a single call to zclient. For example, to generate a text listing of the spot diagram, use the command

```
PostRequestMessage("GetTextFile, \"C:\\OUTPUT.TXT\", Spt, , 0", szBuffer);
```

The spot diagram data will be placed in the text file "OUTPUT.TXT" in the specified folder. For more information, see "GetTextFile".

String Codes

String codes are 3 character arguments used in the keywords "GETTEXTFILE" and "OPENANALYSISWINDOW", and the extension "GetTextFile".
The string codes are listed below. Note that the codes are case sensitive.

"Code"	DESCRIPTION
File codes	
"New"	NEW_FILE
"Ope"	OPEN_FILE
"Sav"	SAVE_FILE
"Sas"	SAVE_AS
"Bac"	BACKUP_TO_ARCHIVE_FILE
"Res"	RESTORE_FROM_ARCHIVE_FILE
"Ins	INSERT_LENS
"Prf"	PREFERENCES
"Ext"	EXIT
"Upd"	UPDATE
"Upa"	UPDATE_ALL
"Gen"	GENERAL_LENS_DATA

"Fie"	FIELD_DATA
"Wav"	WAVELENGTH_DATA
"Nxc"	NEXT_CONFIGURATION
"Lac"	LAST_CONFIGURATION
"LDE"	LENS_DATA_EDITOR
"MFE"	MERIT_FUNCTION_EDITOR
"MCE"	MULTI_CONFIG_EDITOR
"TDE"	TOLERANCE_DATA_EDITOR
"EDE"	EXTRA_DATA_EDITOR
"NCE"	NON_SEQUENTIAL_EDITOR
"Und"	UNDO
"Red"	REDO
Layout codes	
"Lay"	2D_LAYOUT
"L3d"	3D_LAYOUT
"Lwf"	WIREFRAME_LAYOUT
"Lsm"	SOLID_MODEL_LAYOUT
"Lsh"	SHADED_MODEL_LAYOUT
"Ele"	ZEMAX_ELEMENT_DRAWING
"ISO"	ISO_ELEMENT_DRAWING
"L3n"	NSC_3D_LAYOUT
"LSn"	NSC_SHADED_MODEL_LAYOUT
"Obv"	NSC_OBJECT_VIEWER
"Pvr"	CAD_PART_VIEWER
Fan codes	
"Ray"	RAY_FAN

"Opd"	OPD_FAN
"Pab"	PUPIL_ABERRATION_FAN
Spots codes	
"Spt"	SPOT_DIAGRAM
"Stf"	THROUGH_FOCUS_SPOT
"Sff"	FULL_FIELD_SPOT
"Sma"	SPOT_MATRIX
"Smc"	SPOT_MATRIX_CONFIG
MTF codes	
"Mtf"	MODULATION_TF
"Tfm"	THROUGH_FOCUS_MTF
"Smf"	SURFACE_MTF
"Mth"	MTF_VS_FIELD
"Fmm"	FFT_MTF_MAP
"Gmm"	GEOMETRIC_MTF_MAP
"Hmf"	HUYGENS_MTF
"Hsm"	HUYGENS_SURFACE_MTF
"Hmh"	HUYGENS_MTF_VS_FIELD
"Gtf"	GEOMETRIC_MTF
"Tfg"	THROUGH_FOCUS_GTF
"Htf"	HUYGENS_THROUGH_FOCUS_MTF
"Gvf"	GEOMETRIC_MTF_VS_FIELD
PSF codes	
"Fps"	FFT_PSF
"Pcs"	PSF_CROSS_SECTION
"Lsf"	FFT_LINE/EDGE_SPREAD

"Hps"	HUYGENS_PSF
"Hcs"	HUYGENS_PSF_CROSS_SECTION
Wavefront codes	
"Wfm"	WAVEFRONT_MAP
"Int"	INTERFEROGRAM
"Foa"	FOUCAULT_ANALYSIS
Surface sag and phase codes	
"Srs"	SURFACE_SAG
"Srp"	SURFACE_PHASE
"Scv"	SURFACE_CURVATURE
"Ssc"	SURFACE_SAG_CROSS
"Spc"	SURFACE_PHASE_CROSS
"Scc"	SURFACE_CURVATURE_CROSS
RMS codes	
"Rms"	RMS_VS_FIELD
"Rmw"	RMS_VS_WAVELENGTH
"Rmf"	RMS_VS_FOCUS
"Rfm"	RMS_FIELD_MAP
Encircled energy codes	
"Enc"	DIFF_ENCIRCLED_ENERGY
"Gee"	GEOM_ENCIRCLED_ENERGY
"Lin"	LINE/EDGE_SPREAD
"Xse"	EXTENDED_SOURCE_ENCIRCLED
"Rel"	RELATIVE_ILLUMINATION
"Ils"	ILLUMINATION_SCAN
"Ilf"	ILLUMINATION_SURFACE

"Foo"	FOOTPRINT_ANALYSIS
"Uni"	UNIVERSAL_PLOT_1D
"Un2"	UNIVERSAL_PLOT_2D
"Fcd"	FIELD_CURV/DISTORTION
"Grd"	GRID_DISTORTION
"Lon"	LONGITUDINAL_ABERRATION
"Lat"	LATERAL_COLOR
"Sim"	IMAGE_SIMULATION
"Ima"	GEOMETRIC_IMAGE_ANALYSIS
"Ibm"	GEOMETRIC_BITMAP_IMAGE_ANALYSIS
"Lsa"	LIGHT_SOURCE_ANALYSIS
Other Miscellaneous codes	
"ABg"	ABG_DATA_CATALOG
"Agm"	ATHERMAL_GLASS_MAP
"Bfv"	BEAM_FILE_VIEWER
"C31"	SRC_COLOR_CHART_1931
"C76"	SRC_COLOR_CHART_1976
"Caa"	COATING_ABS_VS_ANGLE
"Car"	CARDINAL_POINTS
"Cas"	COAT_ALL_SURFACES
"Caw"	COATING_ABS_VS_WAVELENGTH
"Cca"	CONVERT_TO_CIRCULAR_APERTURES
"Cda"	COATING_DIA_VS_ANGLE
"Cdw"	COATING_DIA_VS_WAVELENGTH
"Cfa"	CONVERT_TO_FLOATING_APERTURES
"Cfm"	CONVERT_TO_MAXIMUM_APERTURES

"Cfo"	CONVERT_FORMAT
"Cfs"	CHROMATIC_FOCAL_SHIFT
"Cgl"	CONVERT_GLOBAL_TO_LOCAL
"Chk"	SYSTEM_CHECK
"Clg"	CONVERT_LOCAL_TO_GLOBAL
"Cls"	COATING_LIST
"Cna"	COATING_RET_VS_ANGLE
"Cng"	CONVERT_TO_NSC_GROUP
"Cnw"	COATING_RET_VS_WAVELENGTH
"Coa"	CONVERT_ASPHERE
"Con"	CONJUGATE_SURFACE_ANALYSIS
"Cpa"	COATING_PHASE_VS_ANGLE
"Cpw"	COATING_PHASE_VS_WAVELENGTH
"Cra"	COATING_REFL_VS_ANGLE
"Crw"	COATING_REFL_VS_WAVELENGTH
"Csd"	NSC_CONCATENATE_SPECTRAL_SOURCE_FILES
"Csf"	NSC_CONVERT_TO_SPECTRAL_SOURCE_FILE
"Cta"	COATING_TRAN_VS_ANGLE
"Ctw"	COATING_TRAN_VS_WAVELENGTH
"Dbl"	DOUBLE_PASS
"Dim"	PARTIALLY_COHERENT_IMAGE_ANALYSIS
"Dip"	BIOCULAR_DIPVERGENCE/CONVERGENCE
"Dis"	DISPERSION_PLOT
"Drs"	NSC_DOWNLOAD_RADIANT_SOURCE
"Dvl"	DISPERSION_VS_WAVELENGTH_PLOT
"Dvr"	NSC_DETECTOR_VIEWER;

"Eca"	EXPLODE_CAD_ASSEMBLY
"Ecp"	EXPLODE_CREO_PARAMETRIC_ASSEMBLY
"Ect"	EDIT_COATING
"Eec"	EXPORT_ENCRYPTED_COATING
"Eia"	EXPLODE_INVENTOR_ASSEMBLY
"Esa"	EXPLODE_SOLIDWORKS_ASSEMBLY
"Fba"	FIND_BEST_ASPHERE
"Fcl"	FIBER_COUPLING
"Fld"	ADD_FOLD_MIRROR
"Flx"	DEL_FOLD_MIRROR
"Fov"	BIOCULAR_FIELD_OF_VIEW_ANALYSIS
"Fvw"	NSC_ZRD_FLUX_VS_WAVELENGTH
"Gbp"	PARAX_GAUSSIAN_BEAM
"Gbs"	SKEW_GAUSSIAN_BEAM
"Gcp"	GLASS_COMPARE
"Gft"	GLASS_FIT
"Gho"	GHOST_FOCUS
"Gip"	GRIN_PROFILE
"Gla"	GLASS_CATALOG
"Glb"	GLOBAL_OPTIMIZATION
"Gmf"	GENERATE_MAT_FILE
"Gmp"	GLASS_MAP
"Gpr"	GRADIUM_PROFILE
"Grs"	NSC_GENERATE_RADIANT_SOURCE
"Gst"	GLASS_SUBSTITUTION_TEMPLATE
"Ham"	HAMMER_OPTIMIZATION

"Hlp"	HELP
"Igs"	EXPORT_IGES_FILE
"Iht"	ANGLE_VS_IMAGEHEIGHT
"Imv"	IMA/BIM_FILE_VIEWER
"Jbv"	JPG/BMP_FILE_VIEWER
"Len"	LENS_SEARCH
"Lok"	LOCK_ALL_WINDOWS
"Ltr"	NSC_LIGHTNING_TRACE
"Mfl"	MERIT_FUNCTION_LIST
"Mfo"	MAKE_FOCAL
"Opt"	OPTIMIZATION
"Pal"	POWER_FIELD_MAP
"Pat"	NSC_ZRDA_PATH_ANALYSIS
"Pci"	PARTIALLY_COHERENT_IMAGE_ANALYSIS
"Per"	PERFORMANCE_TEST
"Pha"	POL_PHASE_ABERRATION
"Pmp"	POL_PUPIL_MAP
"Pol"	POL_RAY_TRACE
"Pop"	PHYSICAL_OPTICS_PROPAGATION
"Ppm"	POWER_PUPIL_MAP
"Pre"	PRESCRIPTION_DATA
"Ptf"	POL_TRANSMISSION_FAN
"Pzd"	PLAYBACK_ZRD_ON_DETECTORS
"Qad"	QUICK_ADJUST
"Qfo"	QUICK_FOCUS
"Raa"	REMOVE_ALL_APERTURES

"Rcf"	RELOAD_COATING
"Rda"	NSC_REVERSE_RADIANCE_ANALYSIS
"Rdb"	RAY_DATABASE
"Rdw"	NSC_ROADWAY_LIGHTING_ANALYSIS
"Rev"	REVERSE_ELEMENTS
"Rg4"	NEW_REPORT_GRAPHIC_4
"Rg6"	NEW_REPORT_GRAPHIC_6
"Rml"	REFRESH_MACRO_LIST
"Rtc"	RAY_TRACE_CONTROL
"Rtr"	RAY_TRACE
"Rva"	REMOVE_VARIABLES
"Rxl"	REFRESH_EXTENSIONS_LIST
"Sag"	SAG_TABLE
"Sca"	SCALE_LENS
"Sdi"	SEIDEL_DIAGRAM
"Sdv"	SRC_DIRECTIVITY
"Sei"	SEIDEL_COEFFICIENTS
"Sfv"	SCATTER_FUNCTION_VIEWER
"Sld"	SLIDER
"Slm"	STOCK_LENS_MATCHING
"Spj"	SRC_PROJECTION
"Spo"	SRC_POLAR
"Spv"	SCATTER_POLAR_PLOT
"Srv"	SRC_RSM_VIEWER
"Ssg"	SYSTEM_SUMMARY_GRAPHIC
"Ssp"	SRC_SPECTRUM

"Sti"	NSC_CONVERT_SDF_TO_IES
"Sur"	SURFACE_DATA
"Sys"	SYSTEM_DATA
"Tde"	TILT/DECENTER_ELEMENTS
"Tls"	TOLERANCE_LIST
"Tol"	TOLERANCING
"Tpf"	TEST_PLATE_FIT
"Tpl"	TEST_PLATE_LISTS
"Tra"	POL_TRANSMISSION
"Trw"	TRANSMISSION_VS_WAVELENGTH
"Tsm"	TOLERANCE_SUMMARY
"Unl"	UNLOCK_ALL_WINDOWS
"Vig"	VIGNETTING_PLOT
"Vop"	VISUAL_OPTIMIZER
"Vra"	VARIABLE_RAD
"Vth"	VARIABLE_THI
"Xdi"	EXTENDED_DIFFRACTION_IMAGE_ANALYSIS
"Xis"	EXPORT_IGES/STEP/SAT/STL_FILE
"Yni"	YNI_CONTRIBUTIONS
"Yyb"	Y-YBAR
"Zat"	ZERNIKE_ANNULAR_TERMS
"Zbb"	EXPORT_ZEMAX_BLACK_BOX_DATA
"Zex"	ZEMAX_EXTENSIONS
"Zfr"	ZERNIKE_FRINGE_TERMS
"Zpd"	NSC_ZEMAX_PART_DESIGNER
"Zpl"	EDIT/RUN_ZPL_MACROS

"Zst"	ZERNIKE_STANDARD_TERMS
"Zvf"	ZERNIKE_COEFFICIENTS_VS_FIELD

The Help Tab

This tab provides access to OpticStudio's Help features.
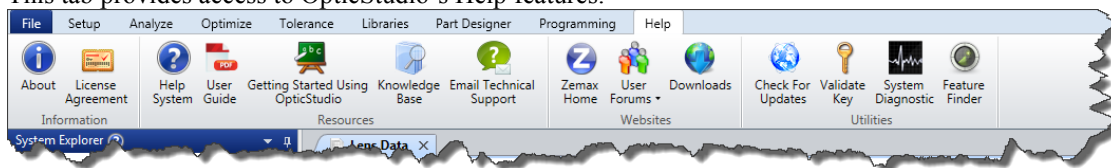
Information Group

This is the Information section of the Help tab.

About

The About button is found in the Information section of the Help tab.

This opens a window with basic information about this program.

Information in the about window includes the license key serial number, the software version, when support expires, and contact information for Radiant Zemax.

License Agreement

The License Agreement button is found in the Information section of the Help tab.

This opens a window with the Zemax the End User License Agreement.
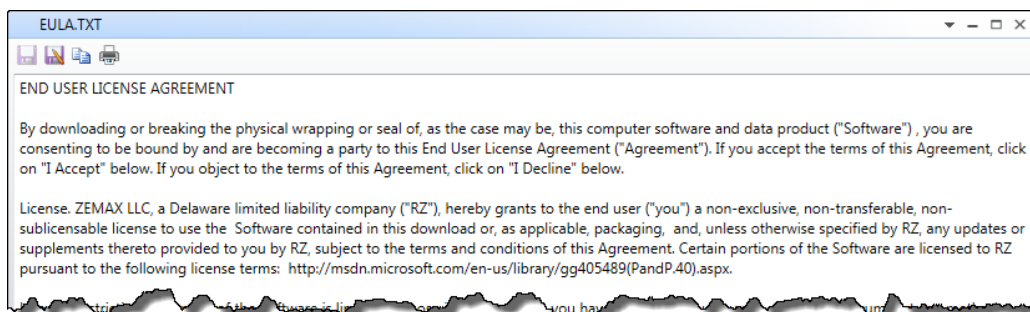
The window with the End User License Agreement includes options to save, copy, or print.

Resources Group

This is the Resources section of the Help tab.
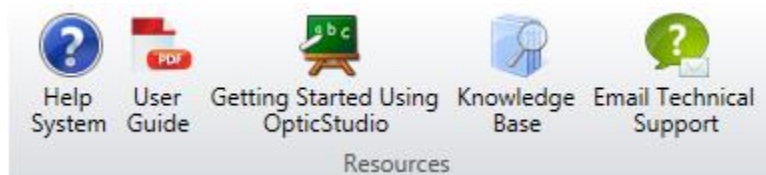
Help System

The Help button is found in the Resources section of the Help tab.

This opens a window with the built-in OpticStudio help file database.
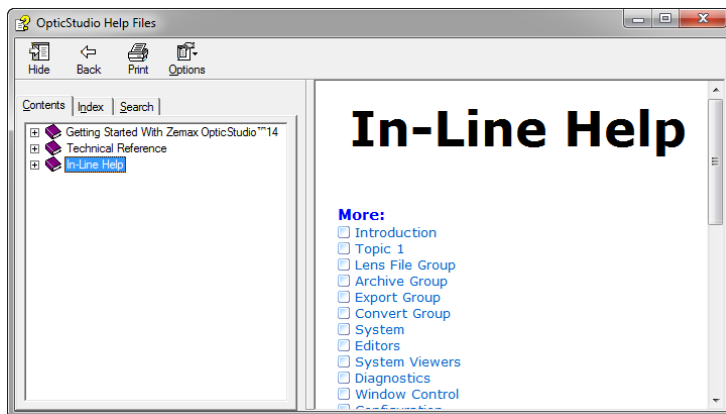
User Guide

The User Guide button is found in the Resources section of the Help tab.
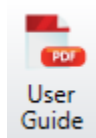
This launches a PDF version of the interactive Help Files.

Getting Started Using OpticStudio

The Getting Started button is found in the Resources section of the Help tab.
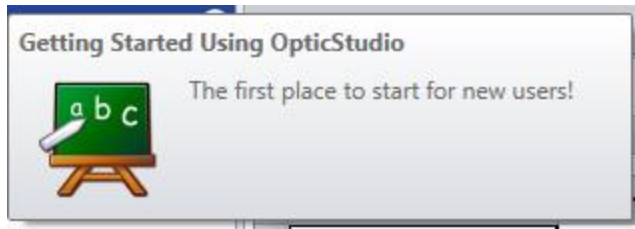
This button opens a window with an introductory section of the OpticStudio help file database.
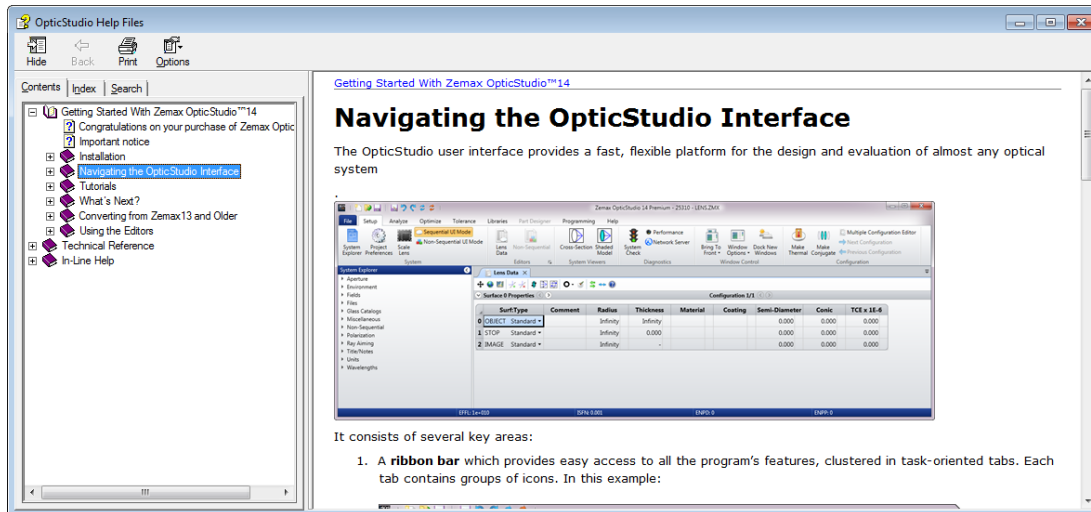
Knowledge Base

The Knowledge Base button is found in the Resources section of the Help tab.
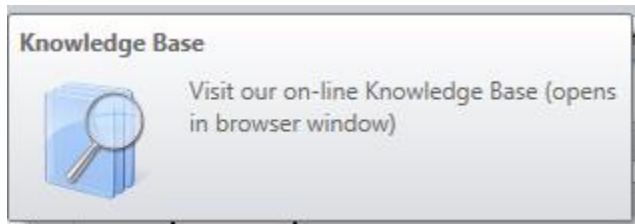
This opens a web browser to the Radiant Zemax on-line Knowledge Base.

Email Technical Support

The Email Technical Support button is found in the Resources section of the Help tab.
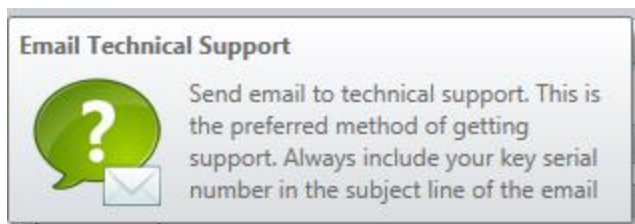
This opens a new email message addressed to Zemax technical support: support@zemax.com.

This is the preferred method of getting support. Always include your key serial number in the subject line of the email.

Websites Group

This is the Websites section of the Help Tab.

Zemax Home

The Zemax Home button is found in the Websites section of the Help tab.
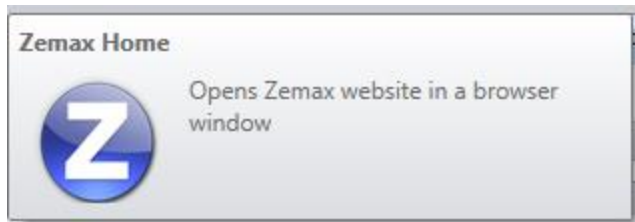
This opens a web browser to the Zemax home page.

The website address is:

http://zemax.com/home

User Forums

The User Forums button is found in the Websites section of the Help tab.
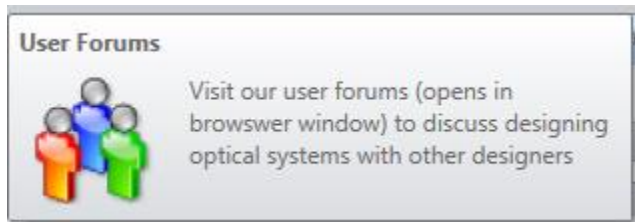
This opens a web browser to the online Zemax user forums in English or Japanese.

The website address is:

http://forum-en.radiantzemax.com/Default.aspx
or
http://www.prolinx.co.jp/zemaxforum/
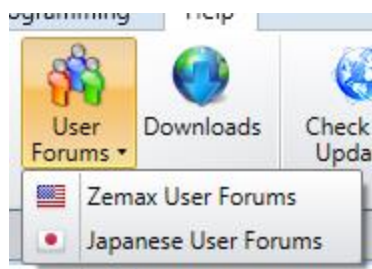
Downloads

The Downloads button is found in the Websites section of the Help tab.
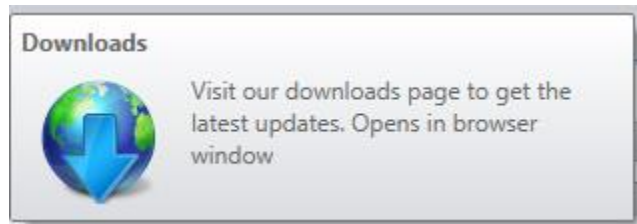
This opens a web browser to our Downloads page to get the latest updates.

The website address is:

http://zemax.com/support/downloads/

Utilities Group

This is the Utilities section of the Help Tab.

Check for Updates

The Check for Updates button is found in the Utilities section of the Help tab.
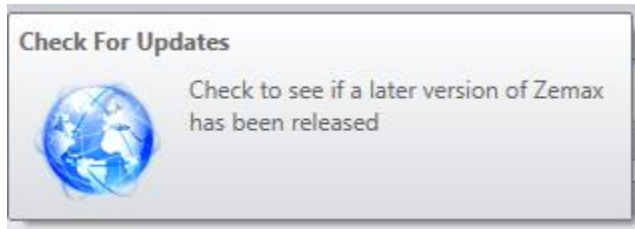
This button checks to see if an updated version of Zemax has been released, and whether or not your Zemax key is elligible to run a newer version:

Or, once you have updated to the latest version:
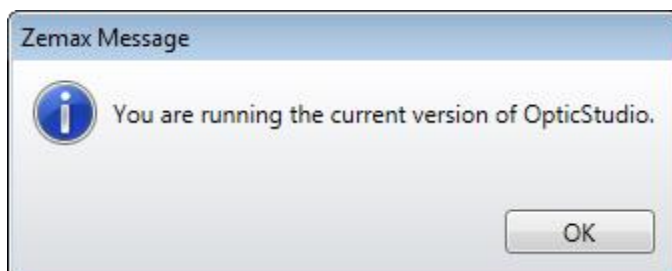
Validate Key

The Validate Key button is found in the Utilities section of the Help tab.
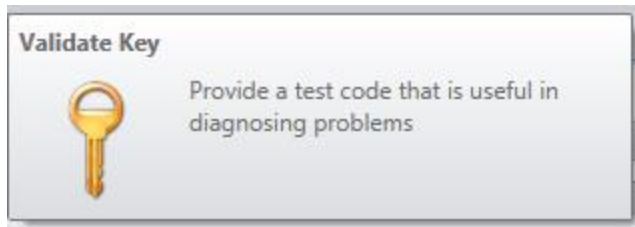
This button provides a test code that is useful is diagnosing problems, and copies the code to the clipboard.
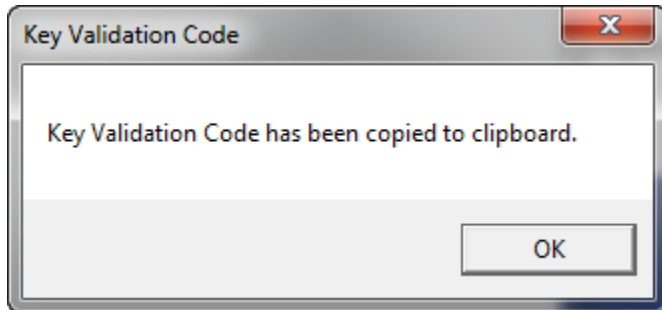
System Diagnostic

The System Diagnostic button is found in the Utilities section of the Help tab.
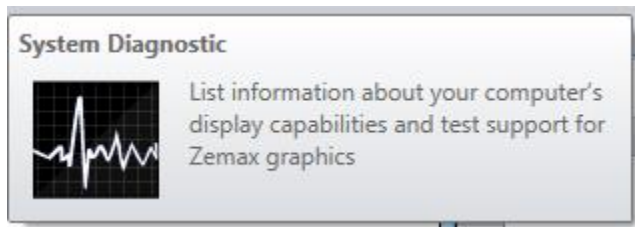
The system diagnostic tool displays graphics rendering information and other diagnostic information specific to your operating system.

The **Save Output** button exports the diagnostic data into a text file.

Here is an example:
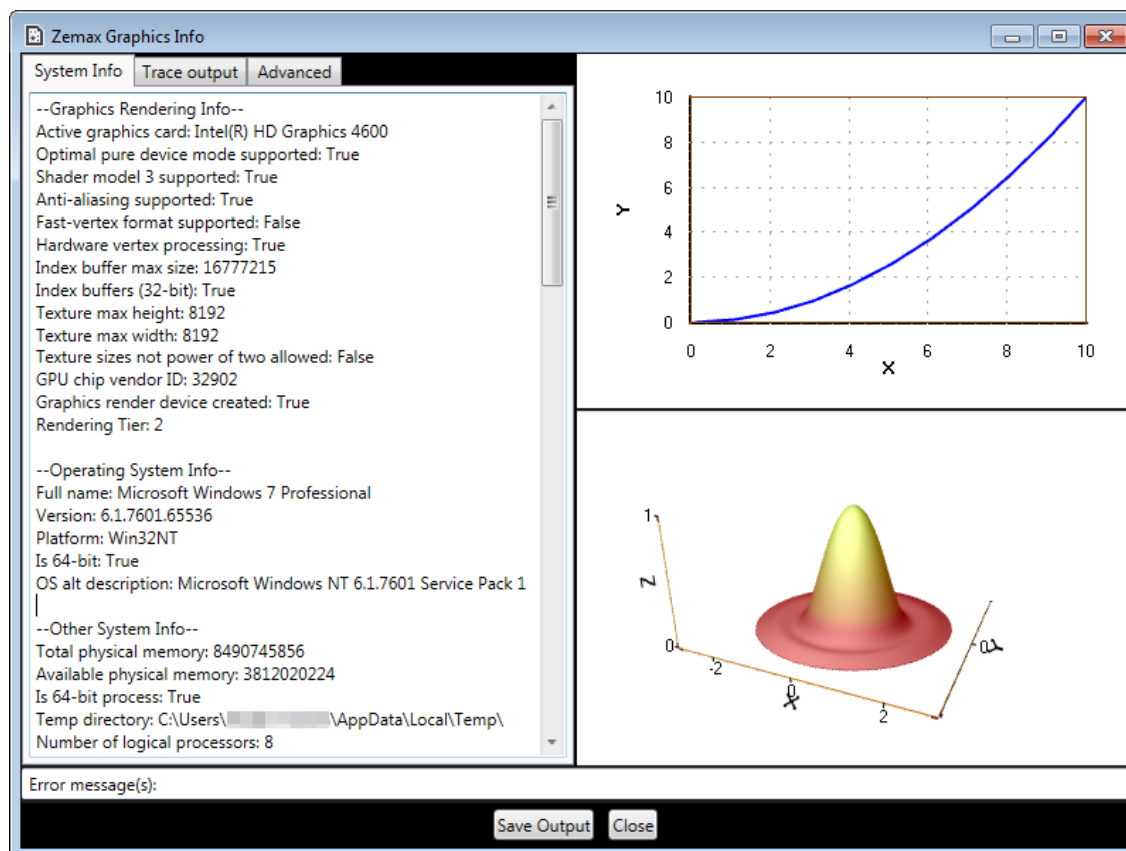
If you are experiencing any graphic display problems, please run this System Diagnostic and email the results to support@zemax.com. In your email, please include:

1. Your Zemax key #

2. A screenshot of the Zemax Graphics Info window

3. Attach the "Save Output" text file

Feature Finder

The Feature Finder button is found in the Utilities section of the Help tab.
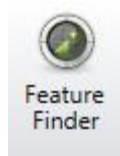
This tool helps users locate different tools and analysis features in the OpticStudio user interface.
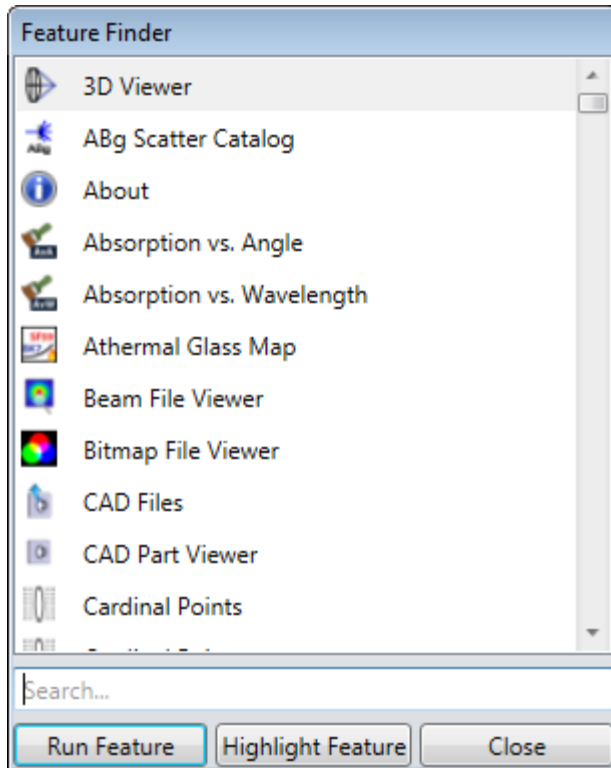
Type a keyword into the search bar to find related analyses or tools.

Button Functions

Run Feature Opens/runs the selected tool or analysis window.

Highlight Feature Highlights the location of the selected tool or analysis in the OpticStudio UI.

Close Closes the Feature Finder window.

Search Bar

The Search Bar is found in the upper right corner of the user interface:
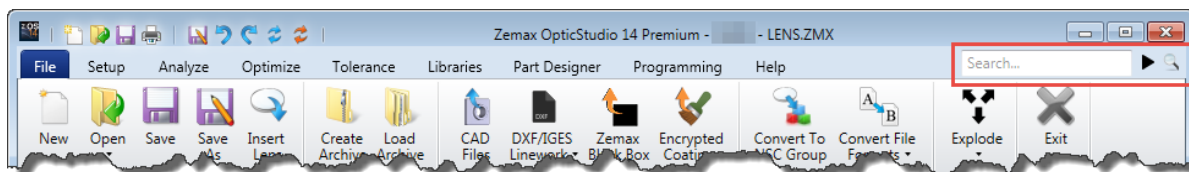
The Search Bar is always visible, regardless of the selected tab, or if the navigation ribbon is hidden. It is similar to the Feature Finder tool, and locates different tools and analysis features in the OpticStudio user interface.

Type a keyword into the search bar to find related analyses or tools.

Button Functions

Run Feature (Play icon) Opens/runs the selected tool or analysis window.

Highlight Feature (Magnifying glass icon) Highlights the location of the selected tool or analysis in the OpticStudio UI.

Index

!

! 686
!, Comment 401

2

2D Analysis Windows 168

3

3D Analysis Windows 169
3D Viewer 939

A

A lens array 409
A pinhole aperture 409
A sample Extension program 1512
A sample Extension program using ZCLIENT 1514
A simplified technique for implementing Extensions 1513
ABCD 417
Aberrations 244, 971
ABg File scattering 515
ABg model scattering 513
ABg Scatter Data Catalogs 1310
Aborting long computations 1169
About 1527
About Extensions 1478
About the ZPL 1369
Absorption vs. Angle 1119, 1196
Absorption vs. Wavelength 1125, 1202
Accounting for polarization 395
Active configuration 739
Add All Data 925
Add Coatings To All Surfaces 845
Add Fold Mirror 840
Adding Coatings to Non-Sequential Objects 126
Adding Coatings to Sequential Surfaces 122
Adding comments to the coating file 226
Adding new source types 632

Adding TCE data 385
Adding thermal index variation data 385
Address 799, 1511
Advanced 788
Algorithm assumptions 408
All other values 384
Alternate Even 417
Alternate methods of defining dispersion data 205
Alternate Odd 418
Amplitude Reflectance, ρ 212
Amplitude Transmittance, τ 212
An Overview of ZPL 1370
Analyze Tab 12
Analyze Tab (Non-Sequential UI Mode) 12
Angle of XY Polarization Ellipse, Ap 214
Angle Scattering 520
Angular magnification 739
Angular spectrum propagation 389
Annular Aspheric Lens 547
Annular Axial Lens 548
Annular Volume 549
Annulus 550
Aperture 775, 851
Apertures 844
APMN, APMX, APTP, APXD, APYD 1392
Apodization 739
Applications Group 1140, 1214
Arc Tool 1365
Archive Group 754
Are There Enough Field Points? 77
Arranging Windows 20
Array 550, 712, 1358
Array Ring 552
Array variables 1372
Arrow Tool 1363
Aspheric Surface 2 555
Assignments 1370
Astigmatic Gaussian 398
Athermal Glass Map 1288
Atmospheric Refraction 418
ATYP, AVAL 1392
Automatic Optimization Group 1227
Automatic Optimization Group (NSC) 1251
Automatic thermal setup 384
Available scatter models 513
Axicon Surface 556

B

Back focal length 739
Beam coordinates and pilot beam properties 405
Beam DLL parameters 401
Beam File Viewer 1092
Beam width and M-squared 406
BEEP 1392
Best Fit Sphere data 301

C

P

PAL/Freeform 1148
Parabolic statistical distribution 682
Parameter 709, 1345
Parameter data 746
Parameter data for sequential surfaces 411
Parameter Solves 835
Paraxial 468, 1428
Paraxial and parabasal rays 746
Paraxial data and ray tracing with NSC 500
Paraxial Gaussian Beam 1094
Paraxial image height 747
Paraxial Lens 602
Paraxial magnification 747
Paraxial working F/# 747
Paraxial XY 469
PARM 1428
Part Designer 1343
Part Designer Tab 15, 1339
Partially Coherent Image Analysis 1075
Path Analysis 1191
Path length through medium, τ 212
PAUSE 1429
Peak Irradiance and total power 405
PENSTYLE 1512
Percent Contribution 238
Performance 929
Performing an optimization 359
Periodic 470
PERTURB 691
PHASE 402
Phase Aberration 1108
Phase Difference Between X and Y, Pxy 214
Phase vs. Angle 1121, 1198
Phase vs. Wavelength 1127, 1204
Phase, P 211
Physical Optics 863
Physical Optics Propagation 387, 1087
Physical Optics Propagation (POP) results 300
Pitfalls when tolerancing 695
Pitfalls with the default merit function 368
PIXEL 1429
Placement Inside, Adjacent, Overlapping 506, 868
Placing sources inside objects 632
Playback ZRD on Detector 1187
PLOT 1429
PLOT2D 1430
Polar Plot 1333
Polarization 785, 1103
Polarization Analysis 209
Polarization and Surface Physics Group 1103
Polarization and thin film coatings 510
Polarization Group 1194
Polarization Pupil Map 1104
Polarization Ray Trace 1103

POLDEFINE 1432
POLTRACE 1432
Polygon 711, 1352
Polygon Object 603
Polynomial 470
POP 1433
Position 714, 1361
Power Field Map 1148
Power Pupil Map 1150
Prescription Data 1131, 1207
Primary wavelength 747
PRINT 1433
PRINTFILE 1434
PRINTWINDOW 1434
Prism/Extrusion 712, 1357
Program Mode 817
Programming Tab 15
Project Preferences 799, 1342
Propagating in and out of the Rayleigh range 393
Propagating through non-sequential surfaces 395
Propagation Phase Factors, pc, ps 212
Propagation through arbitrary optical surfaces 395
Properties of uncoated surfaces 214
PSF 1010
PSF/Strehl Ratio data 255
Pupil Aberration 976
Pupil Integration Settings 1231
PushLens 1498
PushLensPermission 1498
Putting it all together 528
PWAV 1434
Pyramid 711, 712, 1353, 1357

Q

Q-Type Asphere 470
Quantitative beam analysis 405
Quick Adjust 1224
Quick Focus 1223
QUICKFOCUS 1434, 1498
QUIET 691

R

RADI 1435
Radial Grating 471
Radial NURBS 472
Radiant Source Models 1318
Radiant Source Model™ Viewer 1320
Radii 747
Radius 824
Radius of curvature (CRVT) values 382
Radius Solves 828
RANDOMIZE 1435
Ray 714, 1362
Ray Aberration 952, 971