

**ZEMAX®**

---

**Optical Design Program**

**User's Guide**

**February 3, 2005**



ZEMAX Development Corporation

[support@zemax.com](mailto:support@zemax.com)

[www.zemax.com](http://www.zemax.com)

---

## ***Important notice***

ZEMAX® is a registered trademark of ZEMAX Development Corporation Copyright © ZEMAX Development Corporation 1990-2005. All rights reserved.

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

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

The material in this manual is copyrighted and may not be reproduced or electronically distributed without written permission from the copyright holder.

ZEMAX Development Corporation (ZDC) 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 ZDC 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 ZDC has been advised of the possibility of such damages arising from any defect or error in this publication or in the Software.

---

---

# TABLE OF CONTENTS

## INTRODUCTION

## Chapter 1

About this document .....	31
What does ZEMAX do? .....	31
What doesn't ZEMAX do? .....	31
Learning to use ZEMAX.....	32
System requirements .....	32
Multiple processor computers .....	32
Installation procedure.....	32
Installing the key .....	32
Running ZEMAX .....	33
Policy on the key.....	33
Definition of support.....	33
Getting technical support .....	33
Policy on bug fixes .....	33

## USER INTERFACE

## Chapter 2

Introduction .....	35
Types of windows .....	35
Main window operations .....	35
Editor windows operations .....	36
Graphic windows operations.....	36
Using the annotation feature .....	38
Using pan and zoom .....	39
Text windows operations .....	39
Dialog operations .....	39
Aborting long computations .....	40
Summary of useful shortcuts .....	40
Using the Windows clipboard.....	42
ZEMAX file types by extension .....	42

## CONVENTIONS AND DEFINITIONS

## Chapter 3

Introduction .....	45
Active configuration.....	45
Angular magnification .....	45
Apodization .....	45
Back focal length.....	45
Cardinal planes .....	45
Chief ray.....	45
Coordinate axes.....	46
Diffraction limited .....	46
Edge thickness.....	46
Effective focal length.....	47
Entrance pupil diameter .....	47
Entrance pupil position.....	47
Exit pupil diameter .....	47
Exit pupil position .....	47
Extra data.....	47
Field angles and heights .....	47
Float by stop size .....	47

Ghost reflections .....	48
Glasses .....	48
Hexapolar rings .....	48
Image space F/# .....	48
Image space numerical aperture (NA) .....	48
Lens units .....	48
Marginal ray .....	48
Maximum field .....	48
Non-paraxial systems .....	49
Non-sequential ray tracing .....	49
Normalized field and pupil coordinates .....	49
Object space numerical aperture .....	50
Parameter data .....	50
Paraxial and parabasal rays .....	50
Paraxial image height .....	51
Paraxial magnification .....	51
Paraxial working F/# .....	51
Primary wavelength .....	51
Radii .....	51
Real propagation .....	51
Sagittal and Tangential .....	51
Semi-diameters .....	52
Sequential ray tracing .....	52
Strehl ratio .....	52
Surface apertures .....	52
System aperture .....	53
Tangential .....	53
Thicknesses .....	53
Total internal reflection (TIR) .....	53
Total track .....	53
Units .....	53
Vignetting factors .....	53
Virtual propagation .....	55
Wavelength data .....	55
Working F/# .....	55

## FILE MENU

## Chapter 4

New .....	57
Open .....	57
Save .....	57
Save As .....	57
Use Session Files .....	57
Program Mode .....	57
Insert Lens .....	58
Preferences .....	58
Address .....	58
Directories .....	59
Graphics .....	59
Miscellaneous .....	60
Editors .....	61
Printing .....	62
Colors .....	63
Buttons 1-16, Buttons 17-32, Buttons 33-48 .....	63
Status Bar .....	64

Exit.....	64
Recently used files.....	64

## EDITORS MENU

## Chapter 5

Lens Data.....	65
Inserting and deleting surfaces .....	65
The surface number display .....	65
Cutting, copying, and pasting surface data .....	65
Entering surface comments .....	66
Entering radii data .....	66
Entering thickness data .....	66
Entering glass data .....	66
Entering semi-diameter data .....	66
Entering conic data .....	66
Entering parameter data .....	66
The Surface Properties dialog box.....	66
Surface properties type tab .....	67
Surface Type .....	67
Surface DLL .....	67
Surface Color .....	67
Surface Opacity .....	67
Row Color.....	67
Make Surface Stop.....	67
Make Surface Global Coordinate Reference.....	67
Surface Cannot Be Hyperhemispheric .....	67
Surface properties draw tab .....	68
Hide Rays To This Surface .....	68
Skip Rays To This Surface.....	68
Do Not Draw This Surface.....	68
Do Not Draw Edges From This Surface .....	68
Draw Edges As.....	68
Surface properties aperture tab .....	68
Aperture type and other aperture controls.....	68
User defined apertures and obscurations .....	69
The UDA file format.....	70
UDA Examples .....	71
Surface properties scattering tab .....	72
Surface scattering settings .....	73
Surface tilt/decenter tab .....	73
Surface physical optics tab .....	74
Surface coating tab .....	74
Setting and removing solves .....	74
Setting and removing variables.....	74
Menu options.....	74
Edit.....	74
Solves .....	75
Options.....	75
Help.....	75
Merit Function .....	76
Edit.....	76
Tools .....	76
Help.....	77
Multi-Configuration.....	77
Edit.....	77

Solves .....	77
Tools .....	78
Help .....	78
Tolerance Data .....	78
Edit .....	78
Tools .....	79
Help .....	79
Extra Data .....	79
Edit .....	80
Solves .....	80
Tools .....	80
Help .....	80
Non-Sequential Components .....	80
Menu options .....	80
Edit .....	80
Solves .....	81
Errors .....	82
Detectors .....	82
Database .....	82
Help .....	82
Undo, Redo, and Recover .....	82
Undo: None .....	82
Undo: Memory 1 Step .....	82
Undo: Disk Multi Step .....	82

## SYSTEM MENU

## Chapter 6

Update .....	85
Update All .....	85
General .....	85
Aperture .....	85
Aperture Type .....	85
Aperture Value .....	85
Apodization Type .....	86
Uniform apodization .....	86
Gaussian apodization .....	86
Cosine cubed apodization .....	86
User defined apodization .....	86
Apodization Factor .....	87
Telecentric Object Space .....	87
Units .....	87
Lens Units .....	87
Source Units .....	87
Analysis Units .....	87
Title/Notes .....	88
Lens Title .....	88
Notes .....	88
Glass Catalogs .....	88
Ray Aiming .....	88
Ray Aiming .....	88
Use Ray Aiming Cache .....	89
Robust Ray Aiming (slow) .....	89
Pupil Shift: X, Y, and Z .....	89
Environment .....	90
Use Temperature, Pressure .....	90

Temperature in degrees C .....	90
Pressure in ATM.....	90
Polarization .....	90
Unpolarized .....	90
Jx, Jy, X-Phase, Y-Phase.....	90
Convert thin film phase to ray equivalent .....	91
Files.....	91
Coating File .....	91
Scatter Profile.....	91
ABg Data File .....	91
GRADIUM Profile .....	91
Miscellaneous .....	91
Reference OPD .....	91
Paraxial Rays .....	92
Fast Asphere Trace.....	92
Fast Semi-Diameters.....	92
Check GRIN Apertures.....	93
Semi Diameter Margin (lens units) .....	93
Semi Diameter Margin % .....	93
Global Coordinate Reference Surface .....	93
Method To Compute F/# .....	94
Non-Sequential .....	94
Maximum Intersections Per Ray .....	94
Maximum Segments Per Ray.....	95
Maximum Nested/Touching Objects .....	95
Minimum Relative Ray Intensity .....	95
Minimum Absolute Ray Intensity.....	95
Glue Distance In Lens Units.....	95
Missed Ray Draw Distance in Lens Units .....	96
Retrace source rays upon file open.....	96
Fields .....	96
Vignetting factors .....	96
Saving and loading field data .....	96
Wavelengths .....	96
Saving and loading wavelength data .....	97
Next Configuration .....	97
Last Configuration.....	97

## ANALYSIS MENU

## Chapter 7

Introduction .....	99
Layout .....	99
2D Layout.....	99
3D Layout.....	100
The orientation indicator .....	101
Ray errors.....	101
Configuration data .....	102
Raylist file format.....	102
Wireframe.....	102
Solid Model .....	103
Shaded Model .....	103
ZEMAX Element Drawing .....	104
Special characters.....	105
ISO Element Drawing.....	105
Summary of ISO 10110 symbols and codes .....	106

Diameters and bevels.....	107
Tolerance data .....	107
NSC 3D Layout .....	107
NSC Shaded Model .....	108
Fans.....	108
Ray Aberration .....	108
Evaluating results at intermediate surfaces.....	109
Optical Path.....	110
Pupil Aberration.....	110
Spot Diagrams .....	111
Standard.....	111
Through Focus .....	113
Full Field.....	113
Matrix .....	113
Configuration Matrix.....	113
MTF.....	114
FFT MTF .....	114
FFT Through Focus MTF .....	115
FFT Surface MTF.....	116
FFT MTF vs. Field.....	116
Comment about vignetting factors.....	117
FFT MTF Map .....	117
Huygens MTF.....	118
Huygens Through Focus MTF .....	119
Huygens Surface MTF .....	119
Geometric MTF .....	119
Geometric Through Focus MTF .....	120
Geometric MTF vs. Field.....	121
Geometric MTF Map .....	121
PSF.....	122
FFT PSF.....	122
Assumptions used in the FFT PSF calculation.....	123
Discussion of the FFT method and sampling issues.....	123
FFT PSF Cross Section .....	125
FFT Line/Edge Spread.....	125
Huygens PSF.....	126
Huygens PSF Cross Section.....	127
Wavefront.....	128
Wavefront Map.....	128
Interferogram.....	128
Foucault Analysis .....	129
Surface.....	130
Surface Sag .....	130
The Contour Format String.....	131
Surface Phase.....	131
RMS.....	132
RMS vs. Field.....	132
Comments about RMS wavefront computations .....	133
RMS vs. Wavelength.....	133
RMS vs. Focus.....	134
Encircled energy .....	135
Diffraction .....	135
Geometric.....	136
Geometric Line/Edge Spread.....	137



Extended Source.....	137
Illumination.....	139
Relative Illumination.....	139
Effective F/# .....	140
Vignetting Plot.....	140
Illumination XY Scan .....	140
Illumination 2D Surface .....	142
Image Analysis.....	142
Geometric Image Analysis .....	142
The IMA format.....	144
The BIM format.....	144
How rays are chosen for analysis .....	144
Calculating efficiency of multi-mode fibers .....	145
Text output .....	146
Geometric Bitmap Image Analysis .....	146
Diffraction Image Analysis.....	148
Comment about the coherent optical transfer function.....	151
Extended Diffraction Image Analysis .....	151
IMA/BIM File Viewer.....	153
Biocular Analysis.....	153
Field of View.....	153
Dipvergence/Convergence.....	154
Miscellaneous .....	156
Field Curvature/Distortion .....	156
Grid Distortion .....	158
Footprint Diagram .....	160
Longitudinal Aberration .....	160
Lateral Color.....	161
Y-Ybar Diagram .....	161
Chromatic Focal Shift.....	162
Dispersion Diagram.....	162
Glass Map .....	163
Internal Transmittance vs. Wavelength.....	163
System Summary Graphic .....	163
Aberration coefficients .....	164
Seidel Coefficients .....	164
Zernike Fringe Coefficients .....	165
Strehl ratio approximation .....	167
Zernike Standard Coefficients.....	168
Zernike Annular Coefficients .....	171
Calculations .....	174
Ray Trace.....	174
Fiber Coupling Efficiency .....	174
YNI Contributions .....	176
Sag Table.....	177
Cardinal Points.....	178
Glass and Gradient Index .....	178
Dispersion Diagram.....	178
Glass Map .....	178
Internal Transmittance vs. Wavelength.....	179
Grin Profile .....	179
Gradium™ Profile.....	180
Universal Plot.....	180
New Universal Plot.....	180

Polarization .....	182
Polarization Ray Trace.....	182
Polarization Pupil Map .....	182
Transmission.....	183
Phase Aberration .....	183
Transmission Fan.....	184
Coatings.....	184
Reflection vs. Angle .....	184
Transmission vs. Angle .....	185
Absorption vs. Angle .....	185
Diattenuation vs. Angle .....	185
Phase vs. Angle .....	185
Retardance vs. Angle.....	186
Reflection vs. Wavelength.....	186
Transmission vs. Wavelength .....	186
Absorption vs. Wavelength .....	186
Diattenuation vs. Wavelength .....	187
Phase vs. Wavelength .....	187
Retardance vs. Wavelength .....	187
Physical Optics .....	187
Paraxial Gaussian Beam.....	187
Limitations of the analysis .....	188
Overview of Gaussian beams .....	188
Default beam parameters.....	188
Propagating the embedded beam .....	188
The quality factor.....	189
Interactive analysis.....	189
Skew Gaussian Beam.....	189
Physical Optics Propagation .....	190
Comment about beam projection .....	192
Beam File Viewer.....	193

## TOOLS MENU

## Chapter 8

Optimization .....	195
Optimization .....	195
Global Search .....	195
Hammer Optimization .....	195
Merit Function Listing.....	195
Remove All Variables.....	195
Glass Substitution Template .....	195
Tolerancing .....	196
Tolerancing .....	196
Tolerance Listing.....	196
Tolerance Summary.....	196
Test Plates .....	197
Test Plate Fitting .....	197
Test Plate Lists.....	198
Catalogs.....	198
Glass Catalogs.....	198
Glass Compare .....	199
Lens Catalogs .....	199
Coatings.....	202
Edit Coating File.....	202
Reload Coating File.....	203

Add Coatings to All Surfaces .....	203
Coating Listing .....	203
Scattering.....	203
ABg Scatter Data Catalogs .....	203
Wavelength scaling of ABg data .....	204
Scatter Function Viewer .....	204
Apertures .....	206
Convert Semi-Diameters to Circular Apertures.....	206
Convert Semi-Diameters to Floating Apertures .....	206
Remove All Apertures .....	206
Replace Vignetting With Apertures .....	206
Fold Mirrors.....	207
Add Fold Mirror .....	207
Limitations of reversing surfaces following the fold mirror.....	207
Delete Fold Mirror .....	207
Export Data.....	208
Export IGES/SAT/STEP Solid.....	208
Export IGES Line Work .....	210
Export 2D DXF File .....	211
Miscellaneous .....	212
Reverse Elements.....	212
Tilt/Decenter Elements.....	212
Scale Lens .....	212
Make Focal.....	213
Quick Focus .....	213
Convert to NSC Group.....	214
Replicate Object.....	215
Ghost Focus Generator.....	215
Performance Test.....	217
Lock/Unlock All Windows .....	217
Create Polygon Object .....	217
Slider .....	218

## REPORTS MENU

## Chapter 9

Introduction .....	219
Surface Data.....	219
System Data .....	219
Prescription Data .....	219
Comments on computing element volumes .....	220
Report Graphics 4/6.....	220

## MACROS, EXTENSIONS, AND WINDOWS MENUS

## Chapter 10

Edit/Run ZPL Macros.....	223
Refresh Macro List.....	223
Macro Names.....	223
Extensions .....	223
Refresh Extensions List .....	224
Extension Names.....	224
Listing of open windows.....	224

## SURFACE TYPES

## Chapter 11

Introduction .....	225
Parameter data .....	225

Extra data.....	225
Summary of surface types .....	225
User defined surfaces .....	225
Surface types .....	226
ABCD .....	228
Alternate Even .....	228
Alternate Odd.....	229
Atmospheric Refraction.....	229
Biconic .....	229
Biconic Zernike .....	230
Binary Optic 1 .....	230
Binary optic coefficients sign conventions.....	232
Binary Optic 2 .....	232
Binary optic coefficients sign conventions.....	233
Binary Optic 3 .....	233
Binary optic coefficients sign conventions.....	235
Birefringent In and Birefringent Out .....	235
Defining the index .....	236
Defining the crystal axis orientation .....	236
Determining which ray is traced .....	237
Accounting for phase rotation .....	237
Transmission and other properties of birefringent surfaces .....	237
Sample ZEMAX files .....	237
Conjugate.....	238
Coordinate Break.....	239
Cubic Spline.....	240
Comments about spline surfaces.....	240
Cylinder Fresnel.....	240
Diffraction Grating.....	241
Elliptical Grating 1 .....	242
Elliptical Grating surface shape.....	242
Elliptical Grating 2 .....	243
Even Asphere .....	243
Extended Asphere .....	244
Extended Cubic Spline.....	245
Extended Fresnel.....	245
Extended Odd Asphere.....	246
Extended Polynomial .....	247
Extended Toroidal Grating.....	248
Fresnel .....	249
Generalized Fresnel.....	249
Gradient 1 .....	250
Discussion on maximum step size for GRIN surfaces .....	250
Restrictions on surfaces following GRIN surfaces .....	251
Gradient 2 .....	251
Gradient 3 .....	251
Gradient 4 .....	252
Gradient 5 .....	252
Gradient 6 .....	253
Gradient 7 .....	254
GRADIUM™ .....	255
GRADIUM profile file format.....	255
Gradient 9 .....	257
Gradient 10 .....	258

Grid Phase .....	258
Using the shear distance.....	258
Phase coefficients sign conventions .....	259
Grid Sag.....	259
Importing grid data .....	259
Bicubic spline vs. linear interpolation .....	260
Suggestions for using the Grid Sag surface.....	260
Hologram 1 .....	261
Hologram 2 .....	262
Irregular.....	262
Jones Matrix.....	263
Non-Sequential Components.....	263
Odd Asphere.....	263
Odd Cosine .....	263
Optically Fabricated Hologram.....	264
Defining the substrate shape .....	264
Defining the construction optics .....	265
Optimizing the construction and playback optics .....	265
Selecting the hologram type.....	265
Paraxial .....	266
Paraxial XY .....	267
Periodic.....	267
Polynomial .....	268
Radial Grating .....	268
Radial NURBS .....	269
Retro Reflect.....	270
Standard .....	270
Modeling an ellipse with the standard surface .....	270
Modeling an axicon with the standard surface .....	270
Superconic.....	271
Tilted .....	272
Toroidal .....	272
Toroidal Grating .....	273
Toroidal Hologram .....	273
Toroidal NURBS .....	274
User Defined .....	275
The UDS DLL.....	275
Refractive and reflective UDS DLLs.....	276
Gradient index UDS DLLs.....	277
Diffractive UDS DLLs .....	277
Lenslet arrays using UDS DLLs.....	277
User defined surface apodization using DLLs.....	277
Polarization and coating data using DLLs.....	277
Error handling and UDS .....	278
Sample DLLs.....	278
Variable Line Space Grating.....	282
Zernike Fringe Phase.....	282
Zernike Fringe Phase coefficients sign conventions .....	283
Zernike Fringe Sag .....	283
Zernike Standard Phase .....	284
Zernike Standard Phase coefficients sign conventions.....	285
Zernike Standard Sag .....	285
Zone Plate.....	286

Introduction .....	289
Paraxial data and ray tracing with NSC .....	289
Optimization with NSC .....	290
The two methods of using NSC ray tracing .....	290
NSC with ports .....	290
NSC without ports .....	290
Combining NSC with and without ports.....	290
Overview of NSC ray tracing with ports .....	291
The entry port .....	291
The exit port .....	291
Getting rays in .....	292
Tracing rays within the NSC .....	292
Getting rays out .....	292
Overview of NSC ray tracing without ports .....	292
Getting rays in .....	293
Tracing rays within the NSC .....	293
Detecting rays .....	293
Launching rays for analysis.....	293
NSC Objects .....	293
Annular Aspheric Lens .....	297
Annular Volume.....	297
Annulus .....	298
Aspheric Surface.....	298
Aspheric Surface 2.....	299
Axicon Surface .....	299
Biconic Lens.....	300
Biconic Surface .....	301
Making a hyperhemispheric surface.....	302
Binary 1 .....	302
Binary 2.....	303
Binary 2A.....	303
Cone.....	304
Compound Parabolic Concentrator (CPC) .....	305
Cylinder Pipe.....	305
Cylinder Volume.....	306
Cylinder 2 Pipe.....	306
Cylinder 2 Volume.....	306
Diffraction Grating .....	307
Ellipse.....	307
Elliptical Volume.....	307
Even Asphere Lens.....	308
Extended Polynomial Lens.....	308
Extended Polynomial Surface .....	309
Fresnel 1 .....	309
Fresnel 2 .....	310
Hexagonal Lenslet .....	312
Hologram Lens.....	312
Hologram Surface .....	313
Imported .....	314
Comments about imported objects.....	314
Ray tracing accuracy for imported objects .....	314
Ray tracing speed for imported objects.....	315

Limitations on imported objects .....	315
Jones Matrix.....	315
Lenslet 1.....	315
Lenslet 2.....	317
Micro Electro Mechanical System (MEMS).....	317
Null Object.....	318
Odd Asphere Lens .....	318
Polygon Object.....	319
Rectangular Corner.....	319
Rectangle .....	320
Rectangular Pipe.....	320
Rectangular Pipe Grating.....	320
Rectangular Roof .....	321
Rectangular Torus Surface .....	321
Rectangular Torus Volume .....	321
Rectangular Volume.....	322
Rectangular Volume Grating.....	322
Slide .....	322
Sphere.....	323
Standard Lens.....	323
Standard Surface .....	324
STL Object .....	324
Fast Load/Fast Trace STL objects .....	325
Tabulated Faceted Radial .....	325
Tabulated Faceted Toroid .....	326
Tabulated Fresnel Radial.....	327
Toroidal Lens .....	327
Toroidal Surface.....	327
Torus Surface.....	328
Torus Volume.....	328
Triangular Corner.....	329
Triangle .....	329
User Defined Object.....	329
Sample Object DLLs .....	330
Zernike Surface.....	331
The use of facets .....	331
Detectors.....	332
Detector Rectangle object.....	332
Comments on coherent data computations.....	333
Detector Surface object.....	334
Comments about detector triangles .....	335
Detector Volume object.....	335
Objects as detectors .....	335
Sources.....	336
Placing sources inside objects .....	336
Adding new source types.....	336
Parameters common to all source objects .....	337
Source Diode .....	337
Source DLL .....	338
Source DLL parameters .....	338
Creating a new Source DLL .....	338
Sample Source DLLs .....	339
Source Ellipse .....	339
Source Filament.....	340

Source File .....	340
ASCII vs. binary format files .....	340
Memory requirements for source file objects .....	341
Restrictions on the number of rays selected .....	341
Binary Source File format .....	341
ASCII Source File format .....	342
Intensity normalization in source files .....	342
Source Gaussian .....	342
Source Point .....	342
Source Radial .....	342
Source Ray .....	343
Source Rectangle .....	343
Source Tube .....	343
Source Two Angle .....	343
Source Volume Cylinder .....	343
Source Volume Ellipse .....	344
Source Volume Rectangle .....	344
Object Placement .....	344
The object coordinate system .....	344
Reference objects .....	345
Placing objects inside of, adjacent to, or overlapping other objects .....	345
Glue distance .....	345
Nesting object limits .....	345
Nesting volumes .....	346
Nesting surfaces .....	346
Refraction and reflection from NSC objects .....	346
Diffraction from NSC objects .....	347
The object properties dialog box .....	348
Type tab .....	348
Coating/Scattering tab .....	349
Bulk Scattering tab .....	349
GRIN tab .....	349
Diffraction tab .....	350
Sources tab .....	350
Defining paths using the Consider Objects list .....	351
Defining a Scatter Towards list .....	351
User defined apertures .....	352
Coating and scatter groups .....	352
Polarization and thin film coatings .....	352
Coatings on surfaces in contact .....	352
Scattering .....	352
Fraction to scatter and number of scatter rays .....	352
Scatter models .....	353
No scattering .....	354
Lambertian scattering .....	354
Gaussian scattering .....	354
ABg model scattering .....	354
BSDF properties .....	355
The ABg BSDF model .....	355
Defining ABg data .....	356
Scattering rays using the ABg model .....	356
User defined scattering .....	356
Defining an object to use DLL defined surface scattering .....	357
DLL parameters .....	357



Creating a new DLL.....	357
Sample surface scattering: TwoGaussian.DLL .....	357
Bulk scattering .....	357
No Bulk Scattering .....	357
Angle Scattering.....	358
DLL Defined Scattering .....	358
Defining an object to use DLL defined bulk scattering .....	358
DLL parameters.....	358
Creating a new DLL.....	358
Coherence length modeling.....	359
Defining GRIN media for non-sequential ray tracing .....	359
Defining an object to be of gradient index material.....	359
Discussion on maximum step size for GRIN objects .....	359
GRIN DLL parameters .....	359
Creating a new GRIN DLL .....	359
Sample GRIN DLLs.....	360
Defining DLLs for ray splitting at diffractive surfaces .....	361
Defining an object to use the diffraction DLL .....	361
Diffraction DLL parameters .....	361
Creating a new Diffraction DLL .....	361
Ray splitting .....	362
Ray splitting and polarization .....	362
Putting it all together .....	363
Performing a Monte Carlo ray trace.....	363
The Ray Trace/Detector Control .....	363
Choosing the number of analysis rays.....	365
Lost energy .....	365
Ray database files.....	366
Ray Database Viewer .....	367
The Detector Viewer .....	369
The filter string .....	371
Filter string examples .....	373
Defining Polygon Objects.....	373
The comment symbol: !.....	374
The coating group name symbol: C .....	374
The invisible symbol: I.....	374
The vertex symbol: V .....	374
The triangle symbol: T.....	374
The rectangle symbol: R .....	375
Maximum triangles in the polygon object.....	375
Example POB file .....	375
Defining STL Objects .....	376
Maximum triangles in STL objects .....	376
Example STL files .....	376
Special considerations for faceted objects.....	376
Comments about DLLs .....	377

## SOLVES

## Chapter 13

Introduction .....	379
Curvature: Marginal ray angle.....	380
Curvature: Chief ray angle .....	380
Curvature: Pick up .....	380
Curvature: Marginal ray normal .....	381
Curvature: Chief ray normal.....	381

Curvature: Aplanatic .....	381
Curvature: Element power .....	381
Curvature: Concentric with surface .....	381
Curvature: Concentric with radius .....	381
Curvature: F number .....	381
Thickness: Marginal ray height .....	381
Thickness: Chief ray height .....	382
Thickness: Edge thickness .....	382
Thickness: Pick up .....	382
Thickness: Optical path difference .....	382
Thickness: Position .....	382
Thickness: Compensator .....	382
Thickness: Center of curvature .....	383
Glass: Model .....	383
Glass: Pick up .....	383
Glass: Substitute .....	383
Glass: Offset .....	383
Semi-Diameter: Pick up .....	384
Semi-Diameter: Maximum .....	384
Conic: Pick up .....	384
Parameter: Pick up .....	384
Parameter: Chief ray .....	384
Suggestions for use .....	384

## OPTIMIZATION

## Chapter 14

Introduction .....	385
Selecting variables .....	385
Defining the default merit function .....	385
Selecting the type of optimization .....	385
Physically significant merit functions .....	386
Selecting the pupil integration method .....	387
Rings .....	387
Arms .....	388
Grid .....	388
Delete Vignetted .....	388
Setting thickness boundary values .....	388
Start At .....	389
Assume Axial Symmetry .....	389
Ignore Lateral Color .....	389
Relative X Weight .....	389
Overall Weight .....	389
Pitfalls with the default merit function .....	389
Optimization with apodized beams .....	390
Modifying the merit function .....	390
Int1 and Int2 .....	390
Hx, Hy, Px, Py .....	390
Operand weights less than zero .....	390
Operand weights equal to zero .....	391
Operand weights greater than zero .....	391
Merit function definition .....	391
Optimization operands .....	391
Understanding boundary operands .....	421
Using MTF operands .....	421
Performing an optimization .....	422

Defining complex operands .....	423
Optimizing glass selection .....	424
Using model glasses .....	424
Using glass substitution .....	426
Optimizing zoom and multi-configuration lenses .....	426
Optimizing extra data .....	426
Optimizing tolerance sensitivity.....	426
Optimizing objects in a non-sequential group with sequential rays .....	427
Optimizing with sources and detectors in non-sequential mode.....	427
Comments about random numbers and NSTR.....	428
Pixel numbering for detectors .....	428
Optimizing with the IMAE operand.....	428
Using gradient index operands .....	428
DLTN.....	428
LPTD.....	429
User defined operands.....	429
Optimizing with ZPL macros .....	429
Optimizing with externally compiled programs.....	431
Suggestions for use .....	433
The global optimum .....	433

## GLOBAL OPTIMIZATION

## Chapter 15

Introduction .....	435
Capabilities of ZEMAX.....	435
The Global Search algorithm .....	436
The Hammer algorithm .....	437
Optimizing glass selection .....	437
Using glass substitution .....	437
Restricting selected glasses.....	438
Suggestions for use .....	438
Summary.....	439

## TOLERANCING

## Chapter 16

Introduction .....	441
The basic procedure .....	441
Tolerance operands.....	442
TRAD: Tolerance on radius.....	443
TCUR: Tolerance on curvature .....	444
TFRN: Tolerance on fringes.....	444
TTHI: Tolerance on thickness .....	444
TCON: Tolerance on conic.....	444
TSDX, TSDY: Tolerance on surface decenters .....	445
TSTX, TSTY: Tolerance on surface tilts.....	445
TIRX, TIRY: Tolerance on surface TIR .....	445
TIRR: Tolerance on surface irregularity .....	445
TEXI: Tolerance on surface irregularity using the Fringe Zernike model .....	446
TEZI: Tolerance on surface irregularity using the Standard Zernike model.....	447
TPAR: Tolerance on parameter data .....	448
TEDV: Tolerance on extra data values .....	448
TIND: Tolerance on index .....	448
TABB: Tolerance on Abbe .....	448
TCMU: Tolerance on coating multiplier.....	448
TEDX, TEDY: Tolerance on element decenters .....	448

TETX, TETY, TETZ: Tolerance on element tilts.....	449
TOFF: Tolerance off (can be used for comments) .....	449
TUDX, TUDY, TUTX, TUTY, TUTZ: Tolerance on user defined tilts and decenters.....	449
TNPS, TNPA: Tolerances on non-sequential data.....	449
TMCO: Tolerance on multi-configuration data .....	450
Tolerancing with the Irregular surface type.....	450
Tolerance control operands .....	450
CEDV: Define extra data value compensator .....	451
General comments about min and max values on compensators .....	451
CMCO: Define multi-configuration operand compensator .....	451
COMP: Define compensator .....	451
CPAR: Define parameter compensator.....	451
SAVE: Save sensitivity analysis lenses .....	451
SEED: Seed the random number generator .....	452
STAT: Define statistics.....	452
TWAV: Test wavelength.....	452
Defining default tolerances .....	452
Surface tolerances .....	452
Element tolerances .....	453
Defining compensators .....	453
Preparing the lens for tolerancing .....	454
Performing the tolerance analysis.....	454
How ZEMAX computes the tolerance analysis .....	457
Evaluating compensators.....	458
Sensitivity analysis.....	458
Inverse sensitivity analysis.....	459
Monte Carlo analysis .....	460
Normal statistical distribution .....	460
Uniform statistical distribution .....	460
Parabolic statistical distribution .....	461
User defined statistical distribution.....	461
Discussion of Monte Carlo analysis method .....	462
Nesting rules for Monte Carlo analysis .....	462
Using Tolerance Scripts.....	463
Tolerance Script overview.....	463
The Tolerance Script commands .....	463
! .....	463
CEDV .....	463
CLEARCOMP.....	463
CLOSEFILE.....	464
CMCO .....	464
COMP.....	464
CPAR .....	464
FORMAT .....	464
LOADMERIT .....	464
NOMINAL .....	465
OPENFILE.....	465
OPTIMIZE .....	465
PERTURB .....	465
REPORT .....	466
SAVE .....	466
UPDATE.....	467
Tolerance Script example .....	467
Tolerancing multi-configuration (zoom) lenses .....	467

Tolerancing with solves.....	468
Trouble shooting the tolerance results.....	468
Optimizing for tolerance sensitivity .....	468
Pitfalls when tolerancing .....	468
Summary.....	468

## MULTI-CONFIGURATIONS

## Chapter 17

Introduction .....	471
The first step .....	471
Summary of multi-configuration operands .....	471
Defining the number of configurations .....	474
Defining each configuration .....	475
Adding and removing elements .....	475
Changing configurations .....	475
Optimization with multi-configurations .....	475
Suggestions for organizing multiple-configuration merit functions .....	475
Using solves with multi-configuration data .....	476
MCE Pick up solves .....	476
MCE Thermal pick up solves .....	476

## USING GLASS CATALOGS

## Chapter 18

Introduction .....	477
Specifying which glass catalogs to use.....	477
Editing and reviewing glass catalogs .....	477
Description of catalog data .....	478
Creating a new catalog .....	479
Copying or moving glass catalog files.....	479
The glass dispersion formulas .....	480
The Schott formula.....	480
The Sellmeier 1 formula .....	480
The Sellmeier 2 formula .....	480
The Sellmeier 3 formula .....	480
The Sellmeier 4 formula .....	481
The Sellmeier 5 formula .....	481
The Herzberger formula .....	481
The Conrady formula .....	481
The Handbook of Optics 1 formula .....	481
The Handbook of Optics 2 formula .....	481
The Extended formula.....	482
General comments of using dispersion formulas .....	482
Fitting index data.....	482
Fitting melt data .....	482
Discussion of melt fitting method: .....	483
Defining Transmission Data.....	484
Modeling gases and liquids.....	484
Finding a glass quickly.....	485
Using MIL number glasses .....	485
Using model glasses.....	485
Glass catalog sources.....	486
Obsolete catalog data .....	490

## THERMAL ANALYSIS

## Chapter 19

Introduction .....	491
--------------------	-----

Defining temperature and pressure .....	491
Defining wavelengths.....	491
Index of refraction computation.....	491
Environmental effects on index for gradient index, MIL number, and model glasses .....	493
Defining multiple temperature and pressure values.....	493
Defining which parameters consider thermal effects .....	493
Radius of curvature (CRVT) values .....	494
Thickness (THIC) values.....	494
Parameter values .....	494
Extra data values .....	494
All other values.....	495
Defining multiple environments within a single configuration.....	495
Automatic thermal setup .....	495
Adding TCE data.....	495
Modeling gases and liquids.....	495
Adding thermal index variation data.....	495
Optimizing athermal lenses.....	496
Limitations of thermal analysis.....	496

## POLARIZATION ANALYSIS

## Chapter 20

Introduction .....	497
Review of polarization concepts .....	497
The electric field vector .....	497
Field vs. ray phase conventions.....	498
The polarization ellipse .....	498
Definition of terms .....	498
Intensity, I.....	498
Phase, P.....	498
Path length through medium, .....	499
Internal absorption per lens unit, .....	499
Internal Transmittance, IT .....	499
Propagation Phase Factors, pc, ps .....	499
Amplitude Reflectance, .....	499
Amplitude Transmittance, .....	499
Intensity Reflectance, Rs, Rp.....	499
Intensity Transmittance, Ts, Tp.....	499
Intensity Absorption, As, Ap .....	500
Diattenuation, D.....	500
Coating Phase, Ps, Pp .....	500
Retardance, S .....	500
Phase Difference Between X and Y, Pxy .....	500
Major and Minor semi axis of the Polarization Ellipse, EM, Em .....	500
Angle of XY Polarization Ellipse, Ap.....	500
Properties of uncoated surfaces .....	500
Defining coatings in ZEMAX .....	501
Coating file data syntax.....	501
The MATE data section.....	503
The TAPR data section .....	503
The COAT data section.....	505
Defining replicated groups of coating layers .....	506
Defining simple ideal coatings.....	506
The IDEAL data section .....	506
The IDEAL2 data section .....	507
The TABLE data section .....	507

Adding comments to the coating file .....	508
Limits on the amount of coating data .....	508
Editing the coating file .....	508
Optimizing coatings with ZEMAX.....	508
Default materials and coatings supplied with ZEMAX .....	509
Specifying coatings on surfaces .....	510
What ZEMAX does if no coating is specified .....	510
Defining the incident polarization .....	510
Defining polarizing components.....	511
What ZEMAX can compute using polarization analysis.....	512
Bulk absorption and transmission .....	512
Modeling birefringent materials.....	512
Modeling frustrated total internal reflection .....	512
Limitations of polarization analysis .....	514

## PHYSICAL OPTICS PROPAGATION

## Chapter 21

Introduction .....	515
Support for multiple processors .....	516
Diffraction propagation.....	516
Representation of the electric field.....	516
The Fresnel number.....	516
Near and far field.....	517
Angular spectrum propagation .....	517
Fresnel diffraction.....	519
Selecting the correct propagator .....	520
Fraunhofer diffraction .....	520
The pilot beam .....	520
Sign conventions for phase data.....	521
Propagating in and out of the Rayleigh range.....	522
Separation of X and Y propagation .....	523
Comments about point spacing and sampling .....	524
Propagation through arbitrary optical surfaces .....	524
Propagating through non-sequential surfaces .....	525
Accounting for polarization.....	525
Memory requirements .....	525
Defining the initial beam.....	526
Gaussian Waist .....	526
Gaussian Angle.....	527
Gaussian Size+Angle.....	527
Top Hat .....	527
User defined table .....	528
ZEMAX Beam File (ZBF) binary format.....	528
ZEMAX Beam File (ZBF) ASCII format .....	528
User defined DLL program.....	529
Beam DLL parameters .....	529
Creating a new Beam DLL .....	529
Surface specific settings .....	529
Computing Fiber Coupling .....	530
Where the integral is computed .....	531
Defining the fiber mode .....	531
Decenters and tilts .....	531
Choosing the location for the receiving fiber .....	531
Suggestions for use .....	532
Algorithm assumptions.....	533

Samples .....	533
Free space propagation .....	533
A pinhole aperture .....	534
A lens array .....	534
Talbot imaging.....	534
Fresnel lens.....	534

## ZEMAX PROGRAMMING LANGUAGE

## Chapter 22

Introduction .....	535
Creating ZPL macros .....	535
Running ZPL macros .....	535
Assigning ZPL macros to buttons .....	536
An overview of ZPL.....	536
ZPL variables .....	536
ZPL functions .....	537
Using the FICL() function .....	546
ZPL math operations.....	546
ZPL logical operators .....	546
ZPL string variables and operations .....	547
ZPL string logical operators .....	549
ZPL keywords .....	549
APMN, APMX.....	549
APXD, APYD.....	550
APTP.....	550
ATYP, AVAL.....	551
BEEP.....	552
COAT .....	552
CLOSE .....	552
CLOSEWINDOW .....	552
COLOR .....	553
COMMENT.....	553
CONI .....	553
COPYFILE .....	553
CURV .....	554
DELETE .....	554
DELETECONFIG .....	554
DELETEDFILE.....	554
DELETEDMCO.....	555
DELETEDMFO .....	555
DELETEOBJECT .....	555
EDVA .....	555
END.....	556
EXPORTBMP.....	556
EXPORTCAD.....	556
EXPORTJPG .....	557
EXPORTWMF .....	557
FINDFILE .....	557
FLDX, FLDY, FWGT, FVDX, FVDY, FVCX, FVCY, FVAN .....	558
FOR, NEXT .....	558
FORMAT .....	559
FTYP .....	559
GCRS.....	560
GDATE.....	560
GETEXTRADATA .....	560



GETGLASSDATA .....	560
GETLSF .....	561
GETMTF .....	562
GETPSF .....	562
GETSYSTEMDATA .....	563
GETTEXTFILE .....	564
GETVARDATA.....	565
GETZERNIKE .....	566
GLAS.....	567
GLENSNAME.....	567
GOSUB, SUB, RETURN, and END .....	568
GOTO.....	568
GRAPHICS .....	568
GTEXT .....	569
GTEXTCENT .....	569
GTITLE.....	570
HAMMER .....	570
IF-THEN-ELSE-ENDIF.....	570
IMA.....	571
IMASHOW.....	571
IMASUM.....	572
IMPORT_ED .....	572
INPUT .....	572
INSERT .....	572
INSERTCONFIG .....	573
INSERTMCO.....	573
INSERTMFO.....	573
INSERTOBJECT.....	573
LABEL.....	574
LINE .....	574
LOADCATALOG .....	574
LOADLENS .....	574
LOADMERIT .....	575
LOCKWINDOW.....	575
NEXT.....	575
NSTR .....	575
NUMFIELD.....	576
NUMWAVE .....	576
OPEN.....	576
OPTIMIZE .....	577
OPTRETURN.....	577
OUTPUT .....	577
PARM.....	578
PARn.....	578
PARAXIAL.....	578
PAUSE .....	578
PIXEL.....	579
POLDEFINE.....	579
POLTRACE.....	579
POP.....	580
PRINT .....	580
PRINTFILE.....	581
PRINTWINDOW.....	581
PWAV.....	581

QUICKFOCUS .....	582
RADI.....	582
RANDOMIZE.....	582
RAYTRACE.....	583
RAYTRACEX .....	583
READ .....	584
READNEXT.....	585
READSTRING.....	585
RELOADOBJECTS.....	585
REM, ! .....	586
REMOVEVARIABLES.....	586
RENAMEFILE .....	586
RETURN .....	586
REWIND.....	586
SAVELENS .....	587
SAVEMERIT .....	587
SAVEWINDOW.....	587
SCATTER .....	588
SDIA.....	588
SETAIM.....	588
SETAIMDATA .....	588
SETAPODIZATION.....	589
SETCONFIG .....	589
SETDETECTOR .....	589
SETMCOPERAND.....	590
SETNSCPARAMETER .....	590
SETNSCPOSITION .....	591
SETNSCPROPERTY.....	591
SETOPERAND .....	593
SETSTDD .....	593
SETSURFACEPROPERTY .....	593
SETTEXTSIZE .....	595
SETTITLE .....	595
SETUNITS .....	595
SETVAR.....	596
SETVECSIZE.....	596
SETVIG .....	596
SHOWFILE .....	596
SOLVETYPE.....	597
STOPSURF.....	598
SUB.....	598
SURFTYPE .....	598
TELECENTRIC .....	599
TESTPLATEFIT .....	599
THIC.....	599
TIMER.....	599
UNLOCKWINDOW .....	600
UPDATE.....	600
VEC1, VEC2, VEC3, VEC4.....	600
WAVL, WWGT .....	601
XDIFFIA .....	601
ZBFCLR .....	602
General comments about all ZBF related keywords .....	602
ZBFMULT.....	602

ZBFPROPERTIES .....	602
ZBFRESAMPLE .....	603
ZBFSHOW .....	603
ZBFSUM .....	603
ZBFTILT .....	604
Example 1 .....	604
Example 2 .....	605

## ZEMAX EXTENSIONS

## Chapter 23

Introduction .....	607
Requirements for writing Extensions .....	607
Establishing the link .....	608
Terminating the link.....	608
Extracting data from ZEMAX .....	608
The data items .....	609
CloseUDOData .....	609
DeleteConfig .....	609
DeleteObject .....	609
DeleteSurface .....	609
ExportCAD .....	609
ExportCheck.....	610
FindLabel .....	610
GetAddress .....	610
GetAperture.....	610
GetAspect .....	610
GetBuffer .....	610
GetComment.....	611
GetConfig .....	611
GetDate .....	611
GetExtra .....	611
GetField.....	611
GetFile.....	611
GetFirst .....	611
GetGlass .....	611
GetGlobalMatrix .....	612
GetIndex.....	612
GetLabel.....	612
GetMetaFile.....	612
GetMulticon .....	613
GetName .....	613
GetNSCData .....	613
GetNSCMatrix.....	613
GetNSCObjectData.....	613
GetNSCParameter .....	614
GetNSCPosition .....	614
GetNSCSettings.....	614
GetOperand .....	614
GetPath.....	614
GetPolState.....	614
GetPupil .....	614
GetPolTrace .....	615
GetPolTraceDirect.....	615
GetRefresh.....	615
GetSag .....	616

GetSequence .....	616
GetSerial .....	616
GetSettingsData .....	616
GetSolve .....	616
GetSurfaceData .....	617
GetSurfaceDLL .....	617
GetSurfaceParameter .....	617
GetSystem .....	617
GetSystemAper .....	618
GetTextFile .....	618
GetTol .....	618
GetTrace .....	619
GetTraceDirect .....	619
GetUDOSystem .....	620
GetUpdate .....	620
GetVersion .....	620
GetWave .....	620
ImportExtraData .....	620
InsertConfig .....	620
InsertObject .....	621
InsertSurface .....	621
LoadFile .....	621
LoadMerit .....	621
MakeGraphicWindow .....	621
MakeTextWindow .....	622
NewLens .....	622
OpenWindow .....	622
Optimize .....	623
PushLens .....	623
PushLensPermission .....	623
ReleaseWindow .....	623
SaveFile .....	623
SetAperture .....	624
SetBuffer .....	624
SetConfig .....	624
SetExtra .....	624
SetField .....	624
SetFloat .....	625
SetLabel .....	625
SetMulticon .....	625
SetNSCObjectData .....	625
SetNSCPosition .....	625
SetNSCParameter .....	625
SetNSCSettings .....	625
SetOperand .....	626
SetPolState .....	626
SetSettingsData .....	626
SetSolve .....	626
SetSurfaceData .....	626
SetSurfaceParameter .....	626
SetSystem .....	626
SetSystemAper .....	627
SetUDOData .....	627
SetUDOItem .....	627

SetVig.....	627
SetWave.....	627
WindowMaximize .....	627
WindowMinimize .....	628
WindowRestore.....	628
Tracing large numbers of rays .....	628
Step 1: Placing the ray data in the array .....	628
Mode 0: Similar to GetTrace .....	628
Mode 1: Similar to GetTraceDirect.....	630
Mode 2: Similar to GetPolTrace .....	630
Mode 3: Similar to GetPolTraceDirect.....	631
Mode 5: For tracing non-sequential rays.....	632
Step 2: Pass the array to ZEMAX .....	633
How ZEMAX calls the client.....	634
Generating a text window .....	634
Generating a graphic window .....	635
ADDRESS.....	635
BOX.....	635
DATA.....	635
DATE.....	635
FRAME.....	635
LENSNAME .....	635
LINE .....	636
NOFRAME .....	636
PENSTYLE .....	636
TEXT .....	636
TITLE .....	636
A sample Extension program.....	636
The DDE_DEMO code.....	637
A simplified technique for implementing Extensions.....	637
The ZCLIENT program.....	637
A sample Extension program using ZCLIENT .....	638
Writing Extensions from the PhasePlot template .....	638
Getting analysis data using GetTextFile and GetMetaFile.....	639



## **About this document**

ZEMAX is available in two different editions: ZEMAX-SE (Standard Edition) and ZEMAX-EE (Engineering Edition). This manual covers both editions, however certain features are unique to ZEMAX-EE, as noted in the text. If a feature is available in ZEMAX-EE, but not ZEMAX-SE, the following message will usually be shown in the manual where the feature is described:



***This feature is only available in the EE edition of ZEMAX.***

Note that ZEMAX-EE does everything ZEMAX-SE does, plus additional features. This manual covers both editions of ZEMAX running on Microsoft operating systems. ZEMAX® is a registered trademark of ZEMAX Development Corporation.

## **What does ZEMAX do?**

ZEMAX is a program which can model, analyze, and assist in the design of optical systems. The interface to ZEMAX has been designed to be easy to use, and with a little practice it can allow very rapid interactive design. Most ZEMAX features are accessed by selecting options from either dialog boxes or pull-down menus. Keyboard shortcuts are provided for quickly navigating or bypassing the menu structure. This manual provides explanations of the conventions used in ZEMAX, descriptions of procedures, and a description of available features.

## **What doesn't ZEMAX do?**

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 ON LENS DESIGN

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, Virenda	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 Optical Design	Cambridge University Press

Author	Title	Publisher
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 no 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.

## **Learning to use ZEMAX**

A tutorial that covers the basic use of ZEMAX and other topics is available as a help file. To use the tutorial, install ZEMAX, then choose Help, Tutorial from the main menu. The latest tutorial may be downloaded from our web site, [www.zemax.com](http://www.zemax.com). Short courses which cover beginner to advanced use of ZEMAX are offered throughout the year, please check our web site, [www.zemax.com](http://www.zemax.com), for details.

## **System requirements**

ZEMAX requires the current version of the operating system, 200 Mb (or more) of available hard disk space, a CD-ROM drive, a minimum display resolution of 1024 x 768 pixels, and Internet and e-mail access for technical support and program updates. Either a parallel port or a USB port is also required for the key (see below).

The amount of RAM ZEMAX requires varies substantially depending upon the nature of the optical system being modeled and the type of analysis being done. For conventional imaging systems, 256 Mb of RAM is adequate and should be considered a minimum requirement. For modeling of very complex objects, extensive physical optics, or scattering and illumination analysis, substantially more memory may be required for optimal performance. For these types of analysis, 512 Mb of RAM should be considered the minimum, and better performance will be achieved with 1 Gb or more of RAM. It is possible to perform physical optics analysis with ZEMAX using very large arrays that require substantially more RAM than this, see "Memory requirements" on page 525 for details.

## **Multiple processor computers**

Many ZEMAX features are designed to run on multiple processors simultaneously. If more than one CPU is installed in the computer system, ZEMAX will automatically detect and use the additional processors.

## **Installation procedure**

Before you install ZEMAX, please verify that you have the required hardware, defined in the previous section.

## **Installing the key**

ZEMAX is copy-protected using a hardware device called a key. The key only permits one copy of ZEMAX to run at a time. Attempting to run ZEMAX on more than one computer at a time, or attempting to defeat the intention of the key is a violation of the owner's copyright, and is punishable by severe criminal and civil penalties.

Attach the key supplied with the program on the parallel printer port at the rear of the computer if you have been supplied a parallel port type key, or to the USB port if you have been supplied a USB type key. For parallel port keys, if you do not have a printer, place the key in the port by itself. If you have a printer, disconnect the printer cable, then attach the key to the port, then attach the printer cable to the key. Your printer, and all other computer operations, will be unaffected by the presence of the key. You do not need to hook up a printer to the key, but if you do, the printer may have to be turned on. If you do not have the key installed correctly, ZEMAX will issue an error message and not allow the program to run.



## **Running ZEMAX**

The ZEMAX installation program will create a new program group. To run ZEMAX, double click on the ZEMAX icon within this program group.

## **Policy on the key**

ZEMAX is multiple-use protected by a device called a key. The key must be attached to the computer on which ZEMAX is running. The key prevents more than one copy of ZEMAX from running at one time. The registered user of the ZEMAX software is allowed to install the ZEMAX software on more than one computer, as long as there is no possibility that more than one copy of ZEMAX will be running at one time. For example, the user is allowed to install the ZEMAX software on a computer at work and another computer at home, as long as the two installations of ZEMAX are never used at the same time. The key must be physically moved from one computer to another, so that only one copy of ZEMAX can run at one time.

The key supplied with the ZEMAX software is worth the full purchase price of the software. If the key is lost or stolen, it will not be replaced without payment of the full purchase price. Insure the key as you would any other business or personal asset of comparable value. Defective keys, or keys that become inoperable, will be replaced at no charge if the key has a current support contract. Defective keys must be returned before a replacement key will be provided.



**If the key is lost or stolen, it will not be replaced without payment of the full purchase price.**

## **Definition of support**

The purchase of a new license to use ZEMAX includes a period of one year of support. Additional years of support may be purchased at the time the new license is purchased or at a later date. The support period is associated with a specific ZEMAX key. To "have support" for a given ZEMAX key means the support period for that key has not expired. Support consists of two benefits: technical support and program updates.

Technical support is defined as ZEMAX Development Corporation responding to any question that requires an engineer to resolve. This includes answering telephone calls, e-mail, faxes, or letters regarding how to use ZEMAX, understanding ZEMAX features, problems with your key, program installation, or reporting suspected bugs.

Program updates include new features as they are developed, along with any bug fixes recently incorporated in the program. Updates also include new electronic documentation. Program updates are available for download from the ZEMAX Development Corporation web site, [www.zemax.com](http://www.zemax.com), for all users who have support.

## **Getting technical support**

If you have any questions about installing or using ZEMAX, please try the following suggestions to find the information you require:

- 1) Look in the Table of Contents to see if there is a chapter or section on the subject.
- 2) Check the index (at the back of this document) to see if it is referenced.
- 3) Look for a sample file of the type of lens you are trying to set up, if appropriate.

If you still cannot find the information you require, you may need to call, FAX, e-mail, or write for technical support. The phone numbers and address are on the title page of this manual. If you telephone, try to be in front of your computer when you call. Note that technical support is available only if you have support for the ZEMAX key you are using, as described in the previous section.

## **Policy on bug fixes**

All non-trivial computer programs have bugs. Every effort is made to find and fix all bugs before a new version is released. However, the program is sufficiently complex such that even a talented team of pre-release program testers will not find every bug. ZEMAX Development Corporation therefore may provide periodic bug fix releases.

If you find an apparent bug, please report it to us. Try to isolate the exact actions which cause or lead up to the bug. Try to see if the error is unique to the lens file you are working with. Bugs are usually easier to find if they also appear on the example files provided with the program.

If you find a bug which causes significant degradation of program performance while you have support, the bug fix will be provided free of charge. ZEMAX Development Corporation reserves the right to determine what bugs constitute significant degradation. If you do not have support, you may be required to renew your support before receiving the bug fix; in this case you will receive the latest program version.

Bug fixes may not be provided if the problem is traced to incompatible, non-standard, or insufficient hardware, or if you are using an outdated version of the operating system.

### **Introduction**

This chapter describes the conventions used by the ZEMAX user interface, and describes some keyboard shortcuts for common window operations. ZEMAX is easy to use once you learn the conventions used throughout the program. For step by step sample sessions using ZEMAX, see the tutorial (choose Help, Tutorial from the main ZEMAX menu).

### **Types of windows**

ZEMAX has different types of windows, each of which serves a different purpose. The types of windows are:

**The main window:** This window has a large blank area, with a title bar, menu bar, and toolbar at the top. The commands available on this menu bar apply in general to the current optical system as a whole.

**Editor windows:** There are six different editors: the Lens Data Editor, the Merit Function Editor, the Multi-Configuration Editor, the Tolerance Data editor, and in ZEMAX-EE only, the Extra Data Editor and the Non-Sequential Components Editor.

**Graphic windows:** These windows are used to display graphic data such as layouts, ray fans, and MTF plots.

**Text windows:** Text windows are used to display text data such as prescription data, aberration coefficients, and numerical data.

**Dialogs.** Dialogs are pop-up windows which cannot be resized. Dialogs are used to change options or data such as field angles, wavelengths, apertures, and surface types. They are also used extensively to change options on graphic and text windows, such as for changing the number of rays on a layout plot.

All windows can be moved or resized (except dialogs) using the standard mouse or keyboard commands. If you are not familiar with these procedures, see any good book on using Windows or the Windows documentation.

### **Main window operations**

The main window bar has several menu headings. Most menu headings correspond to a chapter of the same name later in this manual. See the specific chapter for detailed instructions on using the features available from each menu option. These are the menu bar headings:

**File:** Used to open, close, save, and rename (save as) lens files.

**Editors:** Used to invoke (make appear) any of the editor windows.

**System:** Used to define properties of the optical system as a whole.

**Analysis:** Analysis groups together features which do not change the lens data, but compute numerical or graphical results from the lens data. These features include layouts, ray fans, spot diagrams, diffraction calculations, and more.

**Tools:** Tools are features which can change the lens data or perform advanced computations on the system as a whole. These include optimization, tolerancing, test plate fitting, and more.

**Reports:** Reports are used to document a lens design. These features include system data summaries, surface data summaries, and report graphics.

**Macros:** Used to edit and run ZPL macros.

**Extensions:** Provides access to ZEMAX Extensions, which are compiled features added to ZEMAX.

**Window:** Selects from the list of currently open windows which window to move to the front of the display.

**Help:** Provides access to the program documentation.

Most of the commonly used menu options have keyboard equivalents which may be faster to use. For example, typing Ctrl-Q will quit ZEMAX. The keyboard shortcuts are listed along side the menu options.

A handy shortcut key for switching between windows within the main window is Control-Tab. This will advance control to the next window in the list of windows maintained by ZEMAX.

The main window also displays a row of buttons just below the menu bar. This row of buttons is used to quickly select common operations. All of the buttons represent functions available on the menus. The button definitions may be changed on the File, Preferences dialog box, on the button bar tabs. The three letter mnemonics used by the buttons are defined on that dialog box tab. A screen resolution of 1024 x 768 or higher is recommended for displaying buttons.

## **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, first highlight the cell, then press the back space or 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 up or down cursors 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 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 screen 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.

Clicking once in any cell will move the cursor to that cell. Double clicking on a cell will invoke the solve dialog box for that cell if one exists. Right mouse clicking will also invoke the solve dialog box for that cell if one exists.

## **Graphic windows operations**

Graphic windows have the following menu items:

Update: This recomputes the data displayed in the window with the current settings.

Settings: Invokes a dialog box which controls the options for that window.

Print: Prints the contents of the window.

Window: The submenus under Window are:

Annotate: See "Using the annotation feature" on page 38 for details. The items under Annotate are:

Line: Draws a single line on the graphic window.

Arrow: Draws a line segment with a directional arrow at the end point.

Text: Prompts for and then draws text on the graphic window.

Box: Draws a box on the graphic window.

Edit: Allows extended editing of annotations.

Copy Clipboard: Copies the window contents to the Windows clipboard. See "Using the Windows clipboard" on page 42 for details.

Export: Exports the displayed graphic as a Windows Metafile, BMP, or JPG file. JPG files support high, medium, and low quality. Medium quality typically produces acceptable image quality with a significant reduction in file size.

Lock: If 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. See also "Lock/Unlock All Windows" on page 217.

Unlock: Unlocks a window previously locked. See also "Lock/Unlock All Windows" on page 217.

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.

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 tab of the File, Preferences dialog box.

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 tab of the File, Preferences dialog box.

Configuration: Selects either the current or any specific configuration for the data to be displayed. The default is "current" which means the data in the window is displayed for the active configuration. Some analysis windows, such as the 3D layout, Report Graphics, and Spot Diagrams, independently allow selection of one or more configurations to be displayed on the "settings" dialog box. That option, if present on the settings box, will override this menu option, and this option will be grayed out.

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.

Text: Displays a text listing of the data in a new window. Not all graphics windows support this option.

Zoom: Controls the "zooming" in on smaller areas of the graphic. See "Using pan and zoom" below. The submenus under the Zoom menu are:

In: Zooms in by 2X centered on the current center position.

Out: Zooms out by 2X.

Last: Restores the previous zoom setting.

Unzoom: Restores the complete view of the graphic.

There are two mouse shortcuts available when using graphic windows:

Double clicking anywhere in the body of graphic window will update the contents. This is the same as selecting Update.

Clicking with the right mouse button anywhere in the body of a graphic window will invoke the Settings dialog box.

## 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 on the screen.

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 of the screen. 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.

Help: Invokes the help function.

## 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.

To pan, drag the scroll bars on the sides of the window. Panning is only allowed after a graphic has been zoomed.

To restore the graphic to the original size, select Zoom, Unzoom from the Graphic window menu bar.

Keyboard shortcuts are supported when using pan and zoom, see "Summary of useful shortcuts" on page 40 for details.

## **Text windows operations**

Text windows have the following menu items:

Update: This recomputes the data displayed in the window with the current settings.

Settings: Invokes a dialog box which controls the options for that window.

Print: Prints the contents of the window.

Window: There are five submenus under Window:

Copy Clipboard: Copies the window contents to the Windows clipboard. See "Using the Windows clipboard" on page 42 for details.

Save Text: Saves the displayed text data in an ASCII file.

Lock: If 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. See also "Lock/Unlock All Windows" on page 217.

Unlock: Unlocks a window previously locked. See also "Lock/Unlock All Windows" on page 217.

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.

Configuration: Selects either the current or any specific configuration for the data to be displayed. The default is "current" which means the data in the window is displayed for the active configuration.

There are two mouse shortcuts available when using text windows:

Double clicking anywhere in the body of graphic window will update the contents. This is the same as selecting Update.

Clicking with the right mouse button anywhere in the body of a graphic window will invoke the Settings dialog box.

## **Dialog operations**

Most dialog boxes are self explanatory. Usually there are buttons for OK and Cancel which are common in Windows dialog boxes.

Analysis features, such as ray fan plots, have dialog boxes which allow selection of various options. All of these dialog boxes have six buttons:

OK: Causes the window to recompute and redisplay the data with the currently selected options.

Cancel: Reverts to those options selected before the dialog was invoked, and does not recompute the data.

Save: Saves the currently selected options for use as the default settings in the future. See below.

Load: Loads the default values that were previously saved. See below.

Reset: Resets the default values to the "factory default" settings.

Help: Calls the ZEMAX help system. The help page displayed will contain information about the options on the active dialog box.

The Save and Load buttons have a dual functionality. When Save is pressed, the settings are saved for the current lens file, as well as all lenses without specific settings of their own. For example, if lens "A" is loaded, and the number of rays on the layout plot for "A" is set to 15, and then Save is pressed, the new default number of rays for "A" will be 15. Also, any new lenses created, or old lenses without their own settings, will also use 15 as the new default. Now suppose that a subsequent lens "B" is loaded, and the number of rays is changed to 9, and then Save is pressed again. For "B", and all lenses that have no specific settings of their own, 9 will be the new default. However, the original "A" lens will retain the setting of 15 because it has its own private settings.

The Load button works in a similar way. When Load is pressed, ZEMAX checks to see if that specific lens has any previous settings saved. If it does, then those settings are loaded. If not, then ZEMAX loads the last saved settings for all lenses. From the previous example, a new lens "C" will load up the settings of 9 rays, since that was the last setting saved; while lenses "A" and "B" would load up settings of 15 and 9, respectively, because they have private settings.

The Save and Load private settings are stored in a file with the same name as the lens file, except the extension is CFG rather than ZMX. No lens data is stored in the CFG file, just the user defined settings for each analysis feature.

The other options that appear on dialog boxes can be selected using either the keyboard or the mouse. For keyboard control, use the Tab and Shift-Tab key combinations to move from option to option. The space bar can be used to toggle the currently selected checkbox. The cursor keys can be used select items within a drop down list. Pressing the first letter of an entry in a drop down box will also select that entry.

## **Aborting long computations**

Certain ZEMAX tools may require relatively long computation times. For example, the 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.

## **Summary of useful shortcuts**

The following tables summarize useful shortcuts, using both the keyboard and the mouse.

### **ZEMAX SHORTCUTS**

Action	Result
Ctrl + Tab	Moves the input focus from window to window.
Shift + Ctrl + Tab	Moves the input focus from window to window (backwards).



Action	Result
Ctrl + letter	Shortcut for many ZEMAX tools and functions. For example, Ctrl-L invokes the 2D layout plot. All the keyboard shortcuts are listed alongside the menu items.
F1..F10	Function keys are also used as shortcuts for many functions. All the keyboard shortcuts are listed alongside the menu items.
Backspace	When an editor window has the input focus, the highlighted cell can be edited using the backspace key. Once the backspace key is pressed, the mouse and left-right cursors can be used for editing.
Left double click	If the mouse is positioned over any graphic or text window, double clicking will cause the window contents to be recalculated and redrawn. This is the same as selecting Update. For editor windows, invokes the solves dialog box.
Right click	If the mouse is positioned over any graphic or text window, right clicking will invoke the settings box for that window. This is the same as selecting Settings. For editor windows, invokes the solves dialog box.
Tab	Moves to the next cell in the editor windows, or the next field in the dialog boxes.
Shift + Tab	Moves to the previous cell in the editor windows, or the previous field in the dialog boxes.
Home/End	On spreadsheet editors, moves to the top left/bottom left of the current editor. On text windows, moves to the top/bottom of the window. On graphic windows, Zoom in/out.
Ctrl + Home/End	On spreadsheet editors, moves to the top right/bottom right of the current editor. On graphic windows, Last Zoom/Unzoom.
Cursor Key (Left, Right, Up, Down)	On spreadsheet editors, moves one cell at a time. On 3D graphic windows, rotates the view around X and Y.
Ctrl + Cursor Key (Left, Right, Up, Down)	On spreadsheet editors, moves one screen at a time. On graphic windows, Pan left, right, up, down.
Page Up/Down	On spreadsheet editors, moves up or down one screen at a time. On 3D graphic windows, rotates the view around Z.
Ctrl + Page Up/Down	On spreadsheet editors, moves to top or bottom of column.

## WINDOWS SHORTCUTS

Action	Result
Alt + Tab	Toggles between all currently running applications. Extremely useful for quickly moving between ZEMAX and other running applications.
Ctrl + Esc	Invokes the Windows task list. The task list allows selection of another running application.
Alt	Selects the top menu bar of the current application.
Alt + letter	Selects the menu option with that letter. For example, Alt + F selects the File menu.
Tab	Moves to the next option or field.
Shift + Tab	Moves to the previous option or field.
Space bar	Toggles selected checkboxes on or off.

Action	Result
Enter	Equivalent to pressing the highlighted or default button in a dialog box.
letter	Pressing the underlined letter of a word in a drop down box will select that entry in the drop down box.

## **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. For example, the graphics within this manual were generated within ZEMAX, copied to the clipboard, and then pasted into a desktop publishing program from the clipboard.

To get ZEMAX graphics and text into the clipboard, choose the graphic or text window desired, then select Window, Copy Clipboard. 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.

Some Windows applications cannot import ZEMAX graphics, even if they appear correctly in the Windows Clipboard Viewer. In this case, the work around is to use the "Export Metafile" option described in "Graphic windows operations" on page 36. Once the metafile is created, most Windows applications can import the graphic.

Another way of getting ZEMAX graphics into other applications is to perform a screen capture which creates a bitmap image of either the entire screen or any single window. To capture the entire screen as a bitmap image, press Ctrl-Print Screen on the keyboard. To capture a single window, select that window and press Alt-Print Screen. Once the screen bitmap has been captured, the image can usually be pasted into another application using Ctrl-V or Edit, Paste, depending upon the application.

## **ZEMAX file types by extension**

The following table lists some of the extensions used by ZEMAX to indicate different types of data files.

### **ZEMAX FILE TYPES BY EXTENSION**

Extension	Description
AGF	ASCII Glass Format. These are ASCII files which contain the data for the glass catalogs. An ASCII 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 control screen.
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.

Extension	Description
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 "session" files, which define the screen configuration 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.
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 an ASCII file which contains the complete description of the lens, including apertures, wavelengths, prescription data, and the merit function.
ZPL	ZEMAX Programming Language macros.



### **Introduction**

This chapter describes 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 Chapter "Multi-Configurations".

### **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 chapter "System Menu".

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 "User Defined" on page 275.

### **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 plane 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 plane 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 plane. 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 plane, 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 uses two different definitions for the term "edge thickness". Usually, the edge thickness is computed for a specific surface by:

$$E_i = Z_{i+1} - Z_i + T_i$$

where  $Z_i$  is the sag of the surface at the +y semi-diameter of the surface,  $Z_{i+1}$  is the sag of the next surface at the +y semi-diameter of the next surface, and  $T_i$  is the axial thickness of the surface. Note that the edge thicknesses are computed accounting for the sag at the respective semi-diameter of each surface, which in general are different.

Note also that edge thickness is normally 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.

The exception to this rule is when computing edge thickness solves. Because the edge thickness solve can change the center thickness, the edge thickness solve can change where rays strike the following surface, which in turn means the semi-diameter of the next surface may change. If the semi-diameter of the next surface is used in the edge thickness computation, an "infinite loop" or circular definition may occur.

For this reason, edge thickness solves compute the edge thickness strictly at the semi-diameter of the first surface, for both surfaces. The semi-diameter of the second surface is never used, although the curvature or shape of the surface is used.

### **Effective focal length**

The distance from the rear principal plane to the paraxial image plane. 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.

### **Extra data**

Extra data values are used to define certain non-standard surface types. For example, extra data values are used to define the phase of diffractive optic surfaces such as the Binary 1 surface type. For a complete discussion of the extra data values see “Extra data” on page 225.

### **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 object coordinates. ZEMAX converts x field angles ( $\alpha_x$ ) and y field angles ( $\alpha_y$ ) 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 chief ray on the 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 chief ray on the image surface.

To set the field type and values, see “Fields” on page 96.

### **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 affect of temperature and pressure on index of refraction data, see "Defining temperature and pressure" on page 491.

## **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 plane.

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 "Field angles and heights" on page 47. To set the field type and values, see "Fields" on page 96.

## **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 (implemented with coordinate break surfaces), 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.

## **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 Sequential ray tracing.

## **Normalized field and pupil coordinates**

Normalized field and pupil coordinates are often used in both the ZEMAX program and documentation. There are four normalized coordinates: Hx, Hy, Px, and Py. The Hx and Hy values are the normalized field coordinates; Px and Py are the normalized pupil coordinates.

The normalized field and pupil coordinates represent points on a unit circle. The radial size of the field is then used to scale the normalized field coordinates, and the entrance pupil radius is used to scale the normalized pupil coordinates. For example, suppose the maximum object height is 10 mm. This would be the case if there are 3 defined object height fields, at 0, 7, and 10 mm. The coordinate (Hx = 0, Hy = 1) would refer to a ray which starts at the top of the object (x = 0 mm, y = 10 mm). The coordinate (Hx = -1, Hy = 0) would refer to a ray which starts at (x = -10 mm, y = 0 mm) on the object surface.

The Hx and Hy values always refer to points on a circle in object angle space whose radius is determined by the maximum radial field. The radial field for each field point individually is determined by computing the square root of the sum of the squared x-field and y-field values. The maximum radial field is the largest of these individual radial fields. For example, if a single field point is defined whose x-field is 10 and y-field is -6, then the maximum radial field is 11.66, and Hx and Hy will then be normalized to this radius. Note that if there are two separate field points, one with an x-field of 10 and a y-field of 0; and a second field point with an x-field of 0 and a y-field of -6, then the maximum radial field is 10, not 11.66. The maximum radial field is computed this way for all field types, whether height or angle. The maximum radial field is always a positive value.

The pupil coordinates work the same way. Suppose the entrance pupil radius (Not diameter) is 8 mm. Then (Px = 0, Py = 1) 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 mm, y = 8 mm).

Note that the normalized coordinates should always be between -1 and 1, and that

$$H_x^2 + H_y^2 \leq 1, P_x^2 + P_y^2 \leq 1.$$

The advantage of using normalized coordinates is that certain rays always have the same coordinates, independent of the size or location of either the object or the entrance pupil. For example, the marginal ray is the ray which travels from the center of the object to the top of the entrance pupil, or (Hx = 0, Hy = 0, Px = 0, Py = 1). The chief ray travels from the top of the field to the center of the pupil, or (0,1,0,0). Note that "top of the field" in

this context means the maximum positive field value in the y direction. If the field type is object height, the maximum y field value is the maximum y coordinate on the object surface. If the field type is angle, then the maximum y field value is the maximum y angle in object space; this ray actually comes from negative coordinate space. For more discussion, see "Field angles and heights" on page 47.

Another significant advantage of this system is that rays defined in normalized coordinates remain meaningful as the pupil size and position changes. Suppose prior to optimizing a lens design, you define a ray set 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.

The normalized field coordinates work even if the field positions are defined in terms of angles. For example, suppose you have selected y-field angles of 0, 7, and 10 degrees. This implies your maximum field "radius" in angular space is 10 degrees. Normalized field coordinates of  $H_x = 0$  and  $H_y = 1$  select the x-field of 0 and the y-field of 10 degrees. The coordinates  $H_x = -0.5$  and  $H_y = 0.4$  select the x-field angle of -5 degrees, and a y-field angle of 4 degrees. Note that even if no x-angles are defined, you may trace rays from these fields using non-zero values for  $H_x$ . Notice that the coordinates are normalized field angles if you define your object in terms of field angles. If you are using object height then the  $H_x$  and  $H_y$  are normalized object heights.

### **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" on page 225.

### **Paraxial and parabal 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 "parabal" 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 parabal 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 paraxial 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 paraxial 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 paraxial 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 plane.

### **Paraxial magnification**

The radial magnification, being the ratio of paraxial image height to object height. The paraxial magnification is measured at the paraxial image plane. 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  $\theta$  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 "Virtual propagation" on page 55 and "Thicknesses" on page 53.

### **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 "Fast Semi-Diameters" on page 92), 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 "Surface properties aperture tab" on page 68.

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 "Fast Semi-Diameters" on page 92 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 Non-sequential ray tracing.

## **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 effect 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 page 68.

## **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.

## **Tangential**

See “Sagittal and Tangential” on page 51.

## **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” on page 239) or surface tilts and decenters (see “Surface tilt/decenter tab” on page 73).

Thicknesses corresponding to real propagation (see “Real propagation” on page 51) 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” on page 55). 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 plane 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 value in non-axial systems.

## **Units**

See “Units” on page 87

## **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 and pupil coordinates” on page 49, 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  $\theta$  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 vignettted! 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 vignettted 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 Chapter "System Menu" 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 "Real propagation" on page 51 and "Thicknesses" on page 53.

## **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 on the "Wavelength Data" dialog box; see the "Wavelengths" on page 96 for details.



***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  $\theta$  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 "Paraxial working F/#" on page 51.

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 "Method To Compute F/#" on page 94.

For a generalized definition of F/# useful for evaluating image brightness, see "Effective F/#" on page 140.





## **New**

### *Purpose:*

Clears the current lens.

### *Discussion:*

This option restores ZEMAX back to the start-up condition. If the current lens has not been saved, ZEMAX will prompt to save the lens before it is destroyed.

## **Open**

### *Purpose:*

Opens an existing ZEMAX lens file.

### *Discussion:*

This option opens a previously saved lens file.

## **Save**

### *Purpose:*

Saves lens file.

### *Discussion:*

This option saves the current lens file. To save the lens under a different name or directory, use Save As.

## **Save As**

### *Purpose:*

Saves lens file, optionally with another name.

### *Discussion:*

This option saves the current lens file with the option to change the current file name or directory.

## **Use Session Files**

### *Purpose:*

Toggles the environment setting between using and not using session files. If the menu option is checked, then session files are used.

### *Discussion:*

For a complete discussion of session files, see "Editors" on page 61.

## **Program Mode**

### *Purpose:*

Selects the user interface and program mode.

### *Discussion:*

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")

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.

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 and Extra Data Editors are 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.

Using Non-Sequential mode makes the operation of the program easier for systems which are non-imaging.

It is always possible to switch from Non-Sequential mode to Sequential mode without loss of data. Switching from Sequential mode to Non-Sequential mode will cause all the sequential data to be deleted.

## **Insert Lens**

### *Purpose:*

Inserts surface data from a previously stored lens data to the current lens file.

### *Discussion:*

This option is similar to the "Open" command, but the current lens data is not disturbed. After the file to insert is selected, ZEMAX 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 by default will ignore the object thickness of the new lens. Therefore, the new lens data to be inserted will start at surface 1 rather than surface 0.

Although this feature can save a lot of typing, the resulting lens may have some extra surfaces, or may need a little hand editing to achieve the desired result.

## **Preferences**

ZEMAX allows users a number of options that can be set and saved so that these options are selected automatically when ZEMAX is run. The main configuration file is ZEMAX.CFG; this file can be deleted to return the start-up configuration to the default state. The preference options are divided into groups as follows.

### **Address**

#### *Purpose:*

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:*

Item	Description
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.

Item	Description
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.

## Directories

### *Purpose:*

These settings determine where ZEMAX will place or search for certain types of files. The "?" button may be used to select the desired path.

### *Settings:*

Item	Description
ZEMAX Path	The directory where ZEMAX will look for files used by the program, such as glass catalogs and stock lens catalogs.
Output Path	The directory where text and graphics output will be sent.
Lens Path	The directory for lens files.
ZPL Path	The directory for ZPL Macros. Macros may also be placed in subdirectories within the specified macro.
Undo Path	The directory for storing lens files for the undo/redo feature.
Stock Path	The directory for the stock lens files.
Objects Path	The directory for Non-Sequential Component object files and DLL's.
Glass Path	The directory for glass catalog files.
Coating Path	The directory for coating definition files.

## Graphics

### *Purpose:*

These settings determine the size, color, and behavior of most ZEMAX graphics windows.

### *Settings:*

Item	Description
B/W Display	By default, ZEMAX draws color graphics to the screen. If this checkbox is selected, then all graphics on screen will be black and white.
B/W Plots	By default, ZEMAX prints color graphics to printers. If this checkbox is selected, then all printed graphics will be black and white. Only printers which support color will print in color, otherwise, gray scales will be used to simulate color.
Show Options First	If this checkbox is selected, then the settings box will appear before any analysis graphic or text is computed and displayed.
Win X,Y Size	The default x and y size for graphics and text windows in pixels. This can be adjusted to suit the monitor size and resolution.
Background	The background color on graphics windows can be selected from this drop down list.

Item	Description
Metafiles	ZEMAX generates several different types of Windows Metafile formats. Metafiles are used to copy graphics to the clipboard, or to copy graphics to files on disk, where the graphics may then be pasted or imported into other Windows programs. Most 16 bit Windows 3.1 applications used the "16 Bit Standard" format. However, some Windows 3.1 applications use a variation called the "16 Bit Placeable" format. Newer, 32 bit applications use the "32 Bit Enhanced" format. When a 32-bit format is used, the extension used is EMF, for enhanced metafile format.
Metafile Pen Width	Determines the pen width in device dependent pixels for graphics which are exported as Metafiles, via either the clipboard or as disk files. Larger values yield thicker lines.
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 screen graphics. Each graphic screen may have it's own aspect ratio modified by using the Window, Aspect Ratio setting.
Use Active Cursor	This setting determines whether or not the active cursor is initially on when creating a new window. Each window may be set to display or not to display the active cursor individually once the window is opened using the Window menu pull down.
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.
Highlight While Scrolling	If selected, then highlighting (see Highlight Layouts above) will occur while scrolling with the vertical scroll bars on the right side of the editors. On complex lens drawings this may interfere with proper updating of the display because the lens drawing updates are not fast enough to keep up with the scroll bar controls.
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.

### Miscellaneous

#### *Purpose:*

Miscellaneous settings.

### *Settings:*

Item	Description
Text Window Font	The font size in points to be used for text displayed in an analysis window, such as a ray fan text listing. To change the size of text displayed in the editors see "Editors" below.
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 <a href="http://www.zemax.com/downloads">www.zemax.com/downloads</a> . An active Internet connection is required for ZEMAX to be able to check for updates. The "Check For Updates" option listed under the Help menu can be used to check for updates anytime.
Remote Desktop	<p>The ZEMAX key driver by default does not allow ZEMAX to run under Window's Remote Desktop feature. To allow ZEMAX to run under Remote Desktop, select this option to "On", then exit and restart ZEMAX. This feature requires version 7.0.0 or later of the Sentinel driver, available from the ZEMAX website <a href="http://www.zemax.com">www.zemax.com</a>. When Remote Desktop is set to "On" only one instance of ZEMAX at a time can run.</p> <p>The reason this is not the default setting is that if ZEMAX terminates abnormally (crashes) the remote desktop enabled driver does not allow ZEMAX to run again for several minutes, until the driver resets itself. Also, the remote driver only allows one application at a time to connect to the key, preventing multiple versions of ZEMAX from running.</p> <p>To modify the setting without running ZEMAX, edit the ASCII text file ZEMAX.CFG found in the same directory as the ZEMAX.exe executable file. Look for the line in the file that starts with RDSK. The syntax is RDSK 0 for Remote Desktop off and RDSK 1 for Remote Desktop on.</p>

### *Editors*

#### *Purpose:*

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.

#### *Settings:*

Item	Description
Decimals	Decimals will alter the number of decimal digits displayed on the lens data editor. 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.
Font Size	The font size in points to be used for text. The default is an 8 point text.
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" is selected from the "System" menu. The "Update" setting will cause an update to be performed whenever new data is typed in to the lens, extra, or multi-configuration editors. "Update All" causes all windows to be updated whenever new data is typed in the editors. See the "System menu" chapter for details on Update and Update All.
Undo	The Undo feature has three selectable states: None, Memory 1 Step, and Disk Multi Step. See the Chapter "Editors Menu" for details on the Undo feature.
LDE Cell Size	The width in characters of a single edit cell in the Lens Data Editor. Larger cells will mean fewer columns can be displayed, but the data is generally more readable.

Item	Description
MFE Cell Size	The width in characters of a single edit cell in the Merit Function Editor.
MCE Cell Size	The width in characters of a single edit cell in the Multi-Configuration Editor.
EDE Cell Size	The width in characters of a single edit cell in the Extra Data Editor.
TDE Cell Size	The width in characters of a single edit cell in the Tolerance Data Editor.
NSC Cell Size	The width in characters of a single edit cell in the Non-Sequential Components Editor.
Default Catalog	The name of the default glass catalog to use. See "Glass Catalogs" on page 88.
Show Comments	If checked, then the surface comments column is displayed in the Lens Data Editor. Otherwise, the column is hidden from view.
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 windows on the screen.
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" on page 223 for more information on extensions.
Color Editor Rows	If checked, then rows in the spreadsheet are color coded. See "Row Color" on page 67.

## Printing

### *Purpose:*

These settings determine characteristics of the printed output.

### *Settings:*

Item	Description
Skip Print Dialog	If this box is checked, then ZEMAX will not present the "Print" dialog box which allows selection of printer type and other options when "Print" is selected from any window. If the box is unchecked, then the default printer settings will be used as specified by the Windows Print Manager.
Rotate Plots	If selected, this will cause all printed graphics to be rotated by 90 degrees. This allows a landscape type orientation while printing in portrait mode. See the discussion below.
Plot Width	See the discussion below.
Pen Width	The width of the pen in pixels. Values of 0 result in fine lines, larger values yield thicker lines.
Print Font Size	The font size in points to be used for text when a text window is printed.
Left Graphic Margin %	The left margin of the plot as a percentage of the maximum possible plot width. Only affects graphics printing.
Right Graphic Margin %	The right margin of the plot as a percentage of the maximum possible plot width. Only affects graphics printing.
Top Graphic Margin %	The top margin of the plot as a percentage of the maximum possible plot height. Only affects graphics printing.
Bottom Graphic Margin %	The bottom margin of the plot as a percentage of the maximum possible plot height. Only affects graphics printing.

Item	Description
Left Text Margin	The left margin to be used when printing text files, measured in average character widths. Note that when printing text files in portrait mode, this is the left margin, allowing room for hole punching and binding of the printed page.
Top Text Margin	The top margin to be used when printing text files, measured in lines. Note that when printing text files in landscape mode, this is the left margin, allowing room for hole punching and binding of the printed page.

#### *Discussion:*

The "Plot Width" control is different from most other settings in ZEMAX, because it is actually used to tell ZEMAX how wide the plots are rather than how wide they should be. Every printer will print ZEMAX graphics at a different size. To get accurate scaling on layout plots and scale bars, ZEMAX must be informed as to how big the plots are when printed. With this information, ZEMAX can print exactly at 1:1, or 2:1, and so on.

Pressing the "Plot Width" button will invoke the Print dialog box. This is the same box that is displayed when printing graphics from ZEMAX. This box allows selection of the printer driver, and usually allows selection of printer specific items such as resolution, orientation, and perhaps other options which vary from printer to printer. Using the Print dialog box, select the driver and mode you normally like to print in. Note ZEMAX prints graphics in "landscape mode" by rotating the plot 90 degrees and then using the portrait setting. This is done because virtually all printers default to portrait mode. It is therefore better to stay in portrait mode all the time, and use the rotate plots feature. ZEMAX will use these settings to determine how wide the actual plot will be when printed, and place this width, in inches, in the "plot width" edit box. Note that the plot width is automatically recalculated for the default printer settings whenever the print orientation or margins are adjusted, or the "Reset" button is pressed.

Once the actual plot width has been computed, the scaling on layout plots will be very accurate. However, it will only be accurate if the same printer driver and mode settings are actually used for printing. If a different printer driver or mode is selected at print time, then the plot width scaling will not automatically be recalculated. To get correct scaling when printing to a new printer or using a different printer mode, the plot width environment variable must be reset using the procedure described above.

Lastly, there are times when it may be required to override the default settings for the plot width. For example, if the final printed output is to be reduced to a certain size for inclusion in another document, then the known final size can be used to get exact scaling at the final image size. To accomplish this, enter the known final image width in inches in the plot width edit box, and press Save. All subsequent printing will be done assuming the final image size specified.

Note that the precise scaling controls specified only affect the Layout and Element Drawing features, since all other plots are scale independent.

### Colors

This dialog box is used to define the colors used for drawing graphics and for the background color of rows in the various spreadsheet editors. 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. The Red, Green, and Blue values define the color. Each entry must be between 0 and 255. Colors are defined using the RGB values for 24 bit color (16 million colors), but will only display up to the resolution provided by the graphics hardware present. The resulting colors are shown to the right of each RGB value. When using the defined colors in the spreadsheet editors, ZEMAX automatically blends the color with white to make the foreground text more readable. The Highlight Color is the color ZEMAX uses to draw lens surfaces and NSC objects currently selected in the LDE and NSCE, respectively.

### Buttons 1-16, Buttons 17-32, Buttons 33-48

#### *Purpose:*

These settings determine which functions are available from the button bar at the top of the ZEMAX main screen. The buttons are only user-definable in sequential mode. In NSC mode, ZEMAX uses a fixed set of button

definitions because many of the sequential button features are unavailable in NSC mode. See “Program Mode” on page 57.

*Discussion:*

There are 48 buttons which can be set to invoke any of the ZEMAX main menu features. Each button has an identical list of menu options which can be associated with the button. Selecting "Off" will cause the button to disappear. To make the interface more manageable, the available buttons are arranged in groups of 16 on separate tabs.

ZPL macros may be assigned to buttons. The name on the button will be the first 3 letters of the macro name. For more information on ZPL macros, see “ZEMAX PROGRAMMING LANGUAGE” on page 535. ZPL macros must be placed in the appropriate directory or sub directory, see “Directories” on page 59.

**Status Bar**

*Purpose:*

These settings determine which values are displayed in the status bar at the bottom of the ZEMAX main screen.

*Discussion:*

There are 4 regions which can be set to display various data, such as EFL, EPD, F/#, etc.

**Exit**

*Purpose:*

Exits ZEMAX.

*Discussion:*

If the lens has been modified, ZEMAX will prompt to save the lens. If not, the program terminates.

**Recently used files**

The most recently used lens files are listed at the bottom of the file menu. Selecting one of these files from the list will cause the file to load. This is a shortcut for using the File, Open command.



## **Lens Data**

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.

## **Inserting and deleting surfaces**

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.

## **The surface number display**

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. Three "special" surfaces, the object, stop, and image, display "OBJ", "STO", and "IMA", respectively, instead of the surface number. 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" on page 66. To define the stop surface, see "Make Surface Stop" on page 67. For information on defining surface aperture data, see "Aperture type and other aperture controls" on page 68. For information on defining tilt/decenter data, see "Surface tilt/decenter tab" on page 73.

## **Cutting, copying, and pasting surface data**

See the discussion on the "Edit" menu below.

## Entering surface comments

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 below.

## Entering radii data

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.

## Entering thickness data

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.

## Entering glass data

The material used for each surface is usually specified by entering the name of the glass in the "Glass" 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 the Chapter "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 after the surface to maintain a constant power at the front and back vertices of the surface. 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 keeps the power at the vertices constant, however, the overall lens power will change slightly due to the change in the optical thickness of the glass. This effect is generally very small for thin lenses.

## Entering semi-diameter data

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" on page 52. For information on vignetting rays with apertures, see "Surface properties aperture tab" on page 68.

## Entering conic data

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" on page 270 for a discussion of how conic surfaces are defined.

## Entering parameter data

Parameter data consists of additional numeric values which define the properties of certain surface types. For more information on parameter data, See the Chapter "Surface Types".

## The Surface Properties dialog box

Double clicking on the left most column (the one with the surface numbers listed) on the surface will invoke the surface properties dialog box. The previous/next surface buttons allow rapid navigation through all surfaces. The following surface properties may be defined on this dialog box.

## Surface properties type tab

### 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 detail in the Chapter "Surface Types". Many optical designs only use the standard surface type.

### Surface DLL

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 the Chapter "Surface Types" under "User Defined" for more information.

### 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 on the Colors tab of the File, Preferences dialog box.

### 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" on page 63. The coloring of rows may be disabled, see "Editors" on page 61. The entire spreadsheet may be set to show default or no colors, see "Menu options" on page 74.

### Make Surface Stop

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" on page 88.

### Make Surface Global Coordinate Reference

The global coordinate reference surface may be any surface in the system. To make the currently selected surface the global coordinate reference surface, choose the "Type" tab and check this box on. See "Global Coordinate Reference Surface" on page 93.

### 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.

## Surface properties draw tab

### 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 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.

## Surface properties aperture tab

### Aperture type and other aperture controls

Surface apertures are used to account for the effects of vignetting. The supported types of surface apertures in ZEMAX are: no aperture, circular aperture, circular obscuration, rectangular aperture, rectangular obscuration, elliptical aperture, elliptical obscuration, spider obscuration, user defined aperture, user defined obscuration, and floating aperture. 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.

Setting aperture data for each surface is done on the surface data dialog box. The surface data dialog box can be reached by double clicking on the left most column in the Lens Data Editor. 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 on the surface data dialog box.

The individual aperture types are described below.

**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:** See the discussion in the next section.

**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.

All of the preceding 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 your situation.

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. It is important to remember that decenters do not move the chief ray; the stop must be on the same axis as the object. To design an off axis telescope, for example, set the stop on axis and decenter the system.

If the identical aperture is used on more than one surface, the aperture "Pick Up From" is a useful feature. Any aperture may be picked up from any previous surface.

### *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 subdirectory. 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 be 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 in ASCII format, 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, which is the default value 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, which is the default value 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 defined 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.

Entity Name/Syntax	Description
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). If a vertex must be defined at (0,0), a work around is to instead use some very small value for one of the points, such as (1e-6, 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 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
```

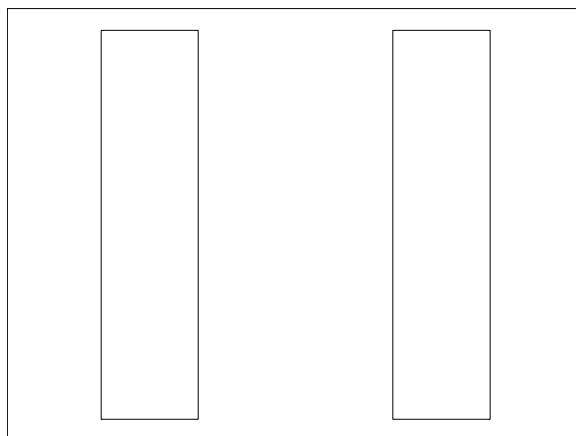
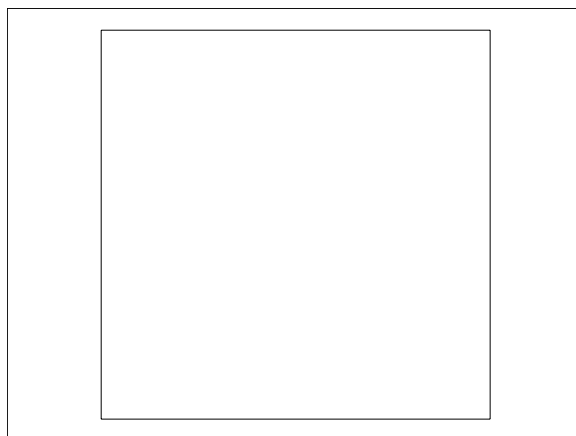
Note the REC command does not require a BRK command to define the end of the aperture.

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
```

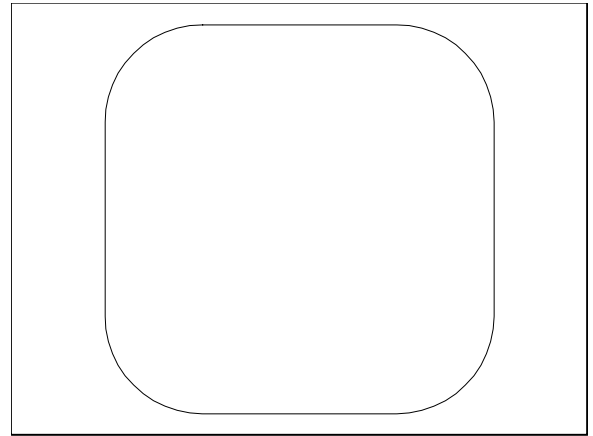
Or the equivalent

```
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
```

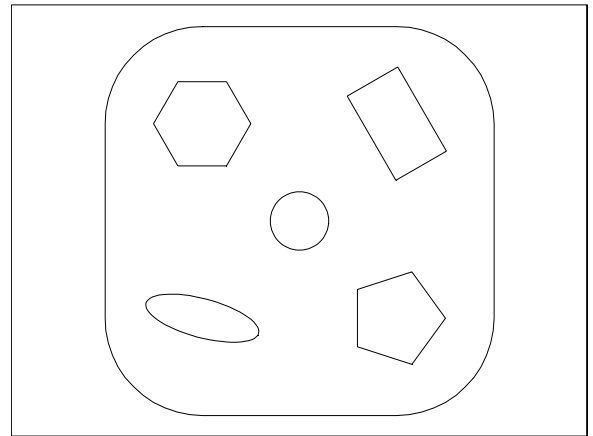


The first LIN command defines the starting point near the upper left corner. The subsequent commands define each of the 8 parts of the aperture. Note a BRK command is not needed if only a single aperture region is defined.

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.

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
```



## Surface properties scattering tab



***This feature is only available in the EE edition of ZEMAX.***

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 the Chapter "Non-Sequential Components" 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.

### Surface scattering settings

ZEMAX uses the same terminology and scattering models for both sequential surfaces and non-sequential components. See the Chapter "Non-Sequential Components" on page 353 for a detailed technical description of the scattering models. For information on defining ABg data, see "ABg Scatter Data Catalogs" on page 203.

The only differences between the non-sequential scattering and the sequential scattering models are:

The non-sequential ABg scattering model supports both reflection and refraction coefficients, while the sequential ABg scattering model uses 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.

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.

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.

### Surface tilt/decenter tab

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" on page 239. A surface tilt/decenter can be thought of as a coordinate break, followed by the surface, followed by another coordinate break.

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.



#### ***The order of the decenters and tilts matters!***

---

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) Pick-up 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) Pick up 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.

For more information on coordinate transformations, see "Coordinate Break" on page 239.

### Surface physical optics tab

See the chapter "PHYSICAL OPTICS PROPAGATION" on page 515 and "Surface specific settings" on page 529 for a description of these features.

### Surface coating tab

The coating tab allows selection of the optical coating to be applied to the surface. For information on defining coatings, see "Defining coatings in ZEMAX" on page 501 and "Optimizing coatings with ZEMAX" on page 508.

This tab supports an additional capability to individually modify the thickness 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. This parameter may be unique for every layer and every surface, even if the same coating is applied to more than one surface. The available controls are:

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.

### Setting and removing solves

Most data columns (such as radii and thickness) support one or more solves. To set a solve on a cell, double click with the left mouse button on that cell. Alternatively, single click with the right mouse button, or select the menu option from the Lens Data Editor menu. Solves are described in "SOLVES" on page 379.

### Setting and removing variables

To make a parameter variable, click on the cell containing the parameter, and press Ctrl-Z on the keyboard. Ctrl-Z also removes the variable status, and acts like a toggle.

### Menu options

The Lens Data Editor menu options are used for inserting and deleting surfaces, selecting surface types, and setting solves and variables.

### Edit

The edit menu offers these options:

Surface Type: This option allows the type of a surface to be changed.

Insert Surface: This inserts a new surface row in the spreadsheet at the current row. The Insert key is a shortcut for this menu option.

Insert After: This inserts a new surface row in the spreadsheet after the current row. The Ctrl-Insert key combination is a shortcut for this menu option.

Delete Surface: This deletes the current row in the spreadsheet. The Delete key is a shortcut for this menu option.

**Cut Surfaces:** Copies all the data for a single surface or a range of surfaces to the Windows Clipboard, then deletes the surfaces. The surface or range of surfaces must first be selected using either of the following techniques:

**Using the mouse:** Click on the first surface to be selected. Hold down the left mouse button, and drag the mouse cursor to cover the range of surfaces desired. The selected surfaces will be highlighted in inverse colors. To select only one surface, drag the mouse up or down from the surface desired until two surfaces are selected, then move the mouse back to the desired surface.

**Using the keyboard:** Move the cursor to any cell on the desired surface. Then hold down the shift key, while moving the cursor up or down until the desired range of surfaces is selected. The selected surfaces will be highlighted in inverse colors. To select only one surface, move the cursor up or down from the surface desired until two surfaces are selected, then move the cursor back to the desired surface.

**Copy Surfaces:** Copies all the data for a single surface or a range of surfaces to the Windows Clipboard. To select a single surface or a range of surfaces, see the discussion under "Cut Surfaces" above. Non-Sequential Component surfaces may not be copied.

**Paste Surfaces:** Copies all the data for a single surface or a range of surfaces from the Windows Clipboard to the current cursor location in the Lens Data Editor. The surface data must first have been copied to the Windows Clipboard using either "Cut Surfaces" or "Copy Surfaces" described above.

**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.

**Edit Cell:** Puts the cell in edit mode.

**Copy Spreadsheet:** Copies either the highlighted range of surfaces, or the entire spreadsheet (if no range of surfaces is selected) to the Windows Clipboard in a text format suitable for being pasted into another Windows application, such as a spreadsheet or word processor. The format is tab delimited text. When using this feature ZEMAX uses curvature rather than radius to retain maximum numerical precision.

**Set Row Colors:** 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, see "Row Color" on page 67.

## Solves

Solves and variables may be placed on many Lens Data Editor values:

**Radius:** Sets solves on the radius of curvature.

**Thickness:** Sets solves on the thickness.

**Glass:** Sets solves on the glass.

**Semi-Diameter:** Sets solves on the semi-diameter.

**Conic:** Sets solves on the conic constant.

**Parameter:** Sets solves on the parameter columns.

**Variable Toggle:** This will toggle the variable status of the currently selected cell. Ctrl-Z is a shortcut for this operation.

## Options

**Show Comments:** If this menu option is checked, then the comment column is displayed. If it is unchecked, then the comment column is hidden. The hide/show status of the comment column is only used for the current session. To set this option automatically at the start of a ZEMAX session, see the Chapter "File Menu" under "Preferences".

## Help

Opens the help system for using the Lens Data Editor.

## **Merit Function**

The Merit Function Editor is used to define, modify, and review the system merit function. The system merit function is used for optimization, and is described in the Chapter "Optimization".

### **Edit**

**Operand Type:** This option allows the type of operand and other data to be changed. For a complete description of merit function operands, see "OPTIMIZATION" on page 385.

**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.

**Delete All:** Deletes all operands in the merit function.

**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 Merit Function Editor. The operand data must first have been copied to the Windows Clipboard using either "Cut Operands" or "Copy Operands" described above.

**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.

**Edit Cell:** Puts the cell in edit mode.

**Copy Spreadsheet:** Copies either the highlighted range of operands, or the entire spreadsheet (if no range of operands is selected) to the Windows Clipboard in a text format suitable for being pasted into another Windows application, such as a spreadsheet or word processor. The format is tab delimited text.

**Set Row Colors:** 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.

### **Tools**

**Update:** This option recomputes the merit function. All operands are evaluated, and the new values are displayed.

**Default Merit Function:** Invokes the dialog box which used to define one of the default merit functions. See the Chapter "Optimization".

**Save:** 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:** 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.

## Help

Opens the help system.

## **Multi-Configuration**

The Multi-Configuration editor is very similar to the Lens Data Editor. To edit the contents of a cell, move the cursor to that cell and type in the new data. To set a solve on a cell, double click with the left mouse button, or select the solve type menu option.

## **Edit**

**Operand Type:** This option allows the type of multi-configuration operand to be changed. For a complete description of multi-configuration operands, see "MULTI-CONFIGURATIONS" on page 471.

**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 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.

**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.

**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.

**Edit Cell:** Puts the cell in edit mode.

**Copy Spreadsheet:** Copies either the highlighted range of operands, or the entire spreadsheet (if no range of operands is selected) to the Windows Clipboard in a text format suitable for being pasted into another Windows application, such as a spreadsheet or word processor. The format is tab delimited text.

**Insert Configuration:** Selecting this item will insert a new column, which corresponds to a new configuration.

**Delete Configuration:** Deletes the configuration indicated by the current position of the cursor. This eliminates the column and all the data it contained.

**Set Row Colors:** 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.

## **Solves**

**Solve Type:** Selecting this option will invoke the solve dialog box for the cell which is currently indicated by the

cursor.

Variable Toggle: This will toggle the variable status of the currently selected cell.

## Tools

**Auto Thermal:** 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" on page 491 and "Limitations of thermal analysis" on page 496.

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 editor 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.



***Check the results of the automatic thermal set-up tool carefully!***

---

**Make Single Config:** This tool will delete all the multi-configuration data, leaving the lens as a single configuration system in whatever configuration was active at the time this tool was selected.

**Add All Data:** This tool will add to the MCE all data currently in the system. 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.

## Help

Opens the help system.

## **Tolerance Data**

The Tolerance Data editor is used to define, modify, and review the system tolerance values. See the Chapter "Tolerancing" for details.

## Edit

**Operand Type:** This option allows the type of operand and other data to be changed. For a complete description of tolerance operands, see "TOLERANCING" on page 441.

**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.

**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.

**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.

**Edit Cell:** Puts the cell in edit mode.

**Copy Spreadsheet:** Copies either the highlighted range of operands, or the entire spreadsheet (if no range of operands is selected) to the Windows Clipboard in a text format suitable for being pasted into another Windows application, such as a spreadsheet or word processor. The format is tab delimited text.

**Set Row Colors:** 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.

## Tools

**Default Tolerances:** Invokes the default tolerances dialog box. See "Defining default tolerances" on page 452 for details.

**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.

**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. See Sort by Surface.

**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.

**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.

## Help

Opens the help system.

## Extra Data



*This feature is only available in the EE edition of ZEMAX.*

---

The extra data editor is very similar to the lens data spreadsheet editor, except only the extra data values are displayed and edited. See “Extra data” on page 225 for a discussion of the surface types which use extra data values. Surfaces cannot be inserted or deleted on the extra data editor.

### Edit

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.

Edit Cell: Puts the cell in edit mode.

### Solves

Solve Type: Selecting this option will invoke the solve dialog box for the cell which is currently indicated by the cursor.

Variable Toggle: This will toggle the variable status of the currently selected cell.

### Tools

Import: The import tool is used to load extra data values for extra data surfaces from an ASCII file rather than by typing the numbers in directly. This menu option will invoke a dialog box which presents a list of ASCII data files which end in the extension .DAT. The dialog box also permits specification of which surface number should receive the data. Numerical data must be in the ASCII file exactly as it appears in the extra data spreadsheet. The format of the ASCII file is a single column of free-format numbers, and the file must end in the DAT extension. ZEMAX will look for the file in the default directory for ZEMAX specified on the Directories tab of the File, Preferences dialog box. 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” on page 259.

### Help

Opens the help system for using the Extra Data Editor.

## **Non-Sequential Components**



***This feature is only available in the EE edition of ZEMAX.***

---

The NSC Editor is only used by the Non-Sequential Components surface type supported by ZEMAX-EE. The editor is very similar to the lens data spreadsheet editor. See “NON-SEQUENTIAL COMPONENTS” on page 289 for a detailed discussion of non-sequential components and ray tracing. 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 Design. For information on changing the program mode, see “Program Mode” on page 57.

### Menu options

The Non-Sequential Components Editor menu options are used for inserting and deleting objects, selecting object types and properties, and setting solves and variables.

### Edit

Object Properties: Activates a dialog box from which the object type, surface and bulk properties may be selected. For complete documentation on the extensive settings available, see “The object properties dialog box” on page 348.

Next Group: If there is more than one Non-Sequential Components surfaces defined in the Lens Data Editor, selecting this option will advance to the group of objects for the next non-sequential surface.

Edit Object: If the object is defined by an ASCII 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.

Reload Object: 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:** This reloads and recreates all objects listed in the editor, see "Reload Object" above.

**Insert Object:** This inserts a new object row in the spreadsheet at the current row. The new object type is "Null Object". The Insert key is a shortcut for this menu option.

**Insert After:** This inserts a new object row in the spreadsheet after the current row. The new object type is "Null Object". The Ctrl-Insert key combination is a shortcut for this menu option.

**Delete Object:** This deletes the current object row in the spreadsheet. The Delete key is a shortcut for this menu option.

**Cut Objects:** Copies all the data for a single object or a range of objects to the Windows Clipboard, then deletes the objects. The object or range of objects must first be selected using either of the following techniques:

Using the mouse: Click on the first object to be selected. Hold down the left mouse button, and drag the mouse cursor to cover the range of objects desired. The selected objects will be highlighted in inverse colors. To select only one object, drag the mouse up or down from the object desired until two objects are selected, then move the mouse back to the desired object.

Using the keyboard: Move the cursor to any cell on the desired object. Then hold down the shift key, while moving the cursor up or down until the desired range of objects is selected. The selected objects will be highlighted in inverse colors. To select only one object, move the cursor up or down from the object desired until two objects are selected, then move the cursor back to the desired object.

**Copy Objects:** Copies all the data for a single object or a range of objects to the Windows Clipboard. To select a single object or a range of objects, see the discussion under "Cut Objects" above.

**Paste Objects:** Copies all the data for a single object or a range of objects from the Windows Clipboard to the current cursor location in the NSC Editor. The object data must first have been copied to the Windows Clipboard using either "Cut Object" or "Copy Objects" described above.

**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.

**Edit Cell:** Puts the cell in edit mode.

**Copy Spreadsheet:** Copies either the highlighted range of objects, or the entire spreadsheet (if no range of objects is selected) to the Windows Clipboard in a text format suitable for being pasted into another Windows application, such as a spreadsheet or word processor. The format is tab delimited text.

**Set Row Colors:** 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 object properties dialog box.

## Solves

Solves and variables may be placed on X, Y, and Z positions, tilts about the X, Y, and Z axis, and any of the parameter data:

**X Position:** Sets fixed, variable, or pickup solves on the X position of an object.

**Y Position:** Sets fixed, variable, or pickup solves on the Y position of an object.

**Z Position:** Sets fixed, variable, or pickup solves on the Z position of an object.

**Tilt about X:** Sets fixed, variable, or pickup solves on the tilt about the X axis of an object.

**Tilt about Y:** Sets fixed, variable, or pickup solves on the tilt about the Y axis of an object.

**Tilt about Z:** Sets fixed, variable, or pickup solves on the tilt about the Z axis of an object.

**Material:** Sets fixed or pickup solves on the material of an object.

**Parameter:** Sets fixed, variable, or pickup solves on any parameter of an 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.

**Variable Toggle:** This will toggle the variable status of the currently selected cell. Ctrl-Z is a shortcut for this operation.

## Errors

Ignore 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.

## Detectors

Ray Trace/Detector Control: Opens up the dialog box which controls clearing the detectors and tracing rays from NSC sources. See the chapter "Non-Sequential Components" for details.

Detector Viewer: Opens up a detector viewer window. A detector viewer can show the data recorded by any defined detector.

## Database

Ray Database Viewer: Opens up a ray database viewer.

## Help

Opens the help system for using the NSC Editor.

## **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 on the Editors tab of the Preferences dialog box described in the Chapter "File Menu".

### 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 from the Editors Menu. 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 directory of Undo files, which by default is a subdirectory within the main ZEMAX directory called \UNDO. 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.



### **Update**

This option only updates data in the Lens Data and Extra Data editors. Update is used to recompute first-order properties, pupil positions, semi-diameters, index data, and solves. Only data presented in the Lens Data and Extra Data editors are affected. See "Update All" in this chapter for more information.

### **Update All**

This option updates all windows to reflect the most recent lens data. ZEMAX does not automatically change all data presented in graphic and text windows to reflect the latest lens data. The reason is that if ZEMAX constantly recalculated MTF, ray fan, spot diagram, and other data while new data is being typed in the lens data editor, then program response would be very slow. Instead, make all the required changes to the lens, and then select Update All to refresh and recompute all the data windows.

Individual graphic and text windows (not editors) can also be updated by double clicking anywhere within the window.

### **General**

This option invokes the General System Data dialog box, which is used to define commonly used data that pertains to the lens as a whole system, rather than data associated with a single surface. The general data is divided into groups as follows.

### **Aperture**

The aperture group consists of the following settings.

#### **Aperture Type**

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. Use the cursor keys to select from the drop down list the desired system aperture type. The system aperture type can be any one of the following:

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 defined conjugate paraxial F/# in image space.

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 normal apodization of rays in the pupil is uniform in angle space, rather than uniform on a plane. This change is required to support angles greater than 90 degrees. This ray distribution may be significantly different than other aperture type settings for large cone angles.

These terms are further defined in the Chapter "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; 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

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 cosine cubed. User defined apodizations are also supported.

#### Uniform apodization

Uniform means rays are distributed uniformly over the entrance pupil, simulating uniform illumination. This is usually the case for distant objects.

#### Gaussian apodization

Gaussian apodization 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  $\rho$  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 apodization

Cosine cubed apodization is appropriate to simulate the intensity fall off characteristic of a point source illuminating a flat plane (such as the entrance pupil, which is always a plane). 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  $\theta$  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

$$A(\rho) = \frac{1}{(1 + (\rho \tan\alpha)^2)^{3/4}},$$

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. Note cosine cubed apodization is only useful for point sources or for field points close to the axis when compared to the entrance pupil diameter.

#### User defined apodization

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 the chapter "Surface Types".

### Apodization Factor

The apodization factor determines how fast the amplitude decays in the pupil. See the previous section on the Apodization Type for details.

### 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.

### Units

#### 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  $\mu\text{m}$ . The term microns is also commonly used in optics. The unit micron is identical to micrometer. Wavelengths are also always defined in micrometers. The choice of lens units has no affect on these units.

#### Source Units

Source units define the unit of measure for the flux (power) or energy emitted by non-sequential sources. This setting only affects sources defined in the non-sequential component editor. 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.

Here is a summary of radiometric, photometric, and energy units by ZEMAX.

#### RADIOMETRIC, PHOTOMETRIC, AND ENERGY UNITS

Radiometric Unit	Photometric Unit	Energy Unit
<u>Flux</u> watt	<u>Flux</u> lumen	<u>joule</u>
<u>Radiant Intensity</u> watt/steradian	<u>Luminous Intensity</u> lumen/steradian = candela	<u>Intensity</u> joule/steradian
<u>Irradiance</u> watt/meter <sup>2</sup> watt/centimeter <sup>2</sup> watt/millimeter <sup>2</sup> watt/foot <sup>2</sup> watt/inch <sup>2</sup>	<u>Illuminance</u> lumen/meter <sup>2</sup> = lux = meter-candle lumen/centimeter <sup>2</sup> = phot lumen/foot <sup>2</sup> = footcandle	<u>Irradiance</u> joule/meter <sup>2</sup> joule/centimeter <sup>2</sup> joule/millimeter <sup>2</sup> joule/foot <sup>2</sup> joule/inch <sup>2</sup>
<u>Radiance</u> watt/steradian-meter <sup>2</sup>	<u>Luminance</u> lumen/steradian-meter <sup>2</sup>	<u>Radiance</u> joule/steradian-meter <sup>2</sup>

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?" on page 31.

## Title/Notes

### Lens Title

The lens title will appear on graphic and text output. The title is specified by typing the title in the required space. Additional text data may be placed on most graphic output, see the Chapter "File Menu" under "Preferences" for details.

### Notes

The notes section allows entry of a few lines of text which is stored with the lens file.

## Glass Catalogs

This control group has an edit field which lists the names of the currently used glass catalogs (without file extensions). Multiple catalogs may be specified by separating the names with a space. To add a catalog to the list, either type the name of the catalog in the list, or choose the catalog name from the drop down list, then click on "Use This Catalog". To remove a catalog from the list, either delete the name of the catalog from the list, or choose the catalog name from the drop down list and click on "Don't Use Catalog".

Glass catalogs must be placed in the directory path \ZEMAX\GLASSCAT, unless the glass catalog directory path has been changed (see "Directories" on page 59). For more information on changing glass catalogs, see "USING GLASS CATALOGS" on page 477. To set a default catalog, see "Editors" on page 61.

## Ray Aiming

The ray aiming group has the following controls.

### Ray Aiming

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 the aberration if ray aiming is selected on. With ray aiming on, every ray trace is performed iteratively, with the program adjusting the ray aiming so that the ray crosses the correct location on the stop surface.

The correct location on the stop surface is determined by first computing the stop surface radius. The correct stop surface coordinates are then computed using a linear scaling of the pupil coordinates. For example, the marginal ray has a normalized pupil coordinate of  $P_y = 1.0$ . The correct coordinate on the stop surface for the marginal ray is the stop surface radius times  $P_y$ .

The stop surface radius is computed by tracing paraxial rays from the center of the object to the stop surface at the primary wavelength. This is the preferred method of computing the stop radius, since paraxial rays are well behaved and paraxial definitions are used for most optical parameters such as focal length, F/#, and magnification. 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 ray system aperture size and



the defined paraxial size. 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. To use real rays instead of paraxial rays to determine the stop radius, choose the "Aberrated" option. Note that for most systems aberrated ray aiming is neither required nor desired. Aberrated 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 aberrated ray aiming causes any of these problems, change back to "On" rather than "Aberrated". Note that once the stop radius is determined, ALL rays, both real and paraxial, are aimed to the correct location on the stop, regardless of whether paraxial or real rays were used to determine the stop radius.

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.

To eliminate any ambiguity in the calculation of the actual stop size, set ray aiming on, 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.

Although 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, ray aiming should only be used when required. To determine the amount of entrance pupil aberration in your system, select ray aiming off, and then look at the pupil aberration plot (see "Pupil Aberration" on page 110). Pupil aberration of less than a few percent is generally insignificant. If your system has significant pupil aberration, select ray aiming on and repeat the calculation. The aberration will decrease to zero, or very nearly so.

### 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 affect 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: X, Y, and Z

For most systems, selecting ray aiming on will eliminate the effects of pupil aberration, as least as far as getting the rays to correctly trace through the system. It does not, of course, actually eliminate the aberration, it merely accounts for it. 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 shifted with respect to the paraxial pupil. This is called "Pupil shift", and there are three components; x, y, and z. The default value of zero for all three may be modified to assist the algorithm in finding a successful first guess for the ray aiming. Positive values for the z shift indicate that the real pupil is behind (to the right of in the conventional optical coordinate system) the paraxial pupil, negative values indicate the pupil is shifted forward. Most wide angle systems have forward-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. The x and y pupil shift values account for the change in location of pupils when the object plane is tilted or when the stop is decentered. If "Scale pupil shift factors by field" is selected, the x and y pupil shift values are also scaled with field, otherwise, the shift value is used for all fields without any scaling. All shifts are in lens units.

It is important to understand that the exact values of the pupil shift is unimportant. Once the first guess ray can be traced, the algorithm will robustly find the exact pupil location. The pupil shift values are just to get the ray aiming started. Generally, guessing at the pupil shift values is an acceptable way of determining a suitable value.

## Environment

### Use Temperature, Pressure

This checkbox should be checked if the temperature and pressure values are different from 20 and 1, respectively. If this box is unchecked, then all temperature and pressure effects are ignored. This speeds up the computation of index data, and so the box should be left unchecked if the nominal temperature and pressure are to be used.

Note that the system wavelengths are referenced to whatever the current system temperature and pressure is.



***Wavelength data are always measured in micrometers referenced to "air" at the current ambient system air pressure.***

---

If the system temperature and air pressure change; care should be taken to adjust the wavelength definitions to match the new environment. See "Wavelength data" on page 55 and "THERMAL ANALYSIS" on page 491 for more information.



***Thermal analysis is only available in the EE edition of ZEMAX.***

---

### Temperature in degrees C

The ambient temperature in degrees Celsius.

### Pressure in ATM

The air pressure in atmospheres. A value of 0.0 implies vacuum, 1.0 implies sea level.

## Polarization

The polarization tab is used to 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 use the "Use Polarization" switch to enable polarization ray tracing and apodization, this tab is the only means for setting the initial polarization state. For these 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 vectorial nature of the electric field is ignored, and scalar theory is still assumed. The rays are attenuated in intensity and a weighted computation is performed. To define the polarization state of non-sequential sources, see "Sources tab" on page 350.



***Polarization ray tracing is only available in the EE edition of ZEMAX.***

---

### 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

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 incident polarization" on page 510.

To edit these values, uncheck the "Unpolarized" box first.

### 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. For more details on the difference between these conventions, see “Field vs. ray phase conventions” on page 498. The recommended and default setting is to convert the field thin film phase to ray phase.

## Files

### Coating File

The name of the file, in the \ZEMAX\COATINGS directory, 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 \ZEMAX\PROFILES directory, 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 “Coating/Scattering tab” on page 349 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 \ZEMAX\ABG\_DATA directory, 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 \ZEMAX\GLASSCAT directory, that contains the GRADIUM surface profile data. The default name is PROFILE.GRD. GRADIUM profile files must end in the extension GRD. See “GRADIUM™” on page 255.

## Miscellaneous

### 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 non zero amplitude. In other words, it is reasonable to assume 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. For more information on diffraction image formation and the importance of the exit pupil, see any of the good books referenced in the first chapter.

As the wavefront propagates from the exit pupil towards the image plane, 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.

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 plane, 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.

However, ZEMAX does permit selection of two other reference methods.

The reference to "Infinity" makes the assumption that the exit pupil is very far away (even if it may not be) and that the OPD correction term is given strictly by the angular error in the ray. There is only one possible reason to use this setting: in the unlikely event that ZEMAX is unable to correctly compute the exit pupil location. This can occur with some very unusual optics which do not form images (real or virtual) of the stop surface. ZEMAX has special code to handle all known cases where this might occur, and therefore, this setting should not be used unless ZEMAX Development Corporation technical support specifically recommends it. There are currently no known cases where this would be the recommended setting.

The reference to "Absolute" means that ZEMAX does not add any correction term at all to the OPD computation, but adds up the total optical path length of the ray and subtracts it from the chief ray. This method is never physically significant, and the only intended purpose is for debugging and testing of the OPD algorithms by ZEMAX Development Corporation.

In summary, always use the "exit pupil" reference, unless specifically instructed to change the setting by a ZEMAX Development Corporation Engineer. Erroneous data may easily result if "exit pupil" is not selected.

### Paraxial Rays

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. Choosing a different setting may result in the failure of ZEMAX to accurately compute all paraxial data, ray aiming, and OPD computations.

There is only one known case where "Consider Coordinate Breaks" may be required. For ray tracing through steeply tilted gratings, coordinate breaks may be required even for paraxial rays, otherwise, the rays may not be able to satisfy the grating equation. This is because diffraction gratings bend rays by an amount that depends critically on the incident angle.

### Fast Asphere Trace

When tracing rays through certain types of aspheric surfaces, iteration is required if no closed form solution exists to the ray-surface intercept equations. If this box is checked (the default condition) ZEMAX makes an initial guess for the ray-surface intercept solution in an attempt to accelerate convergence of the iteration. However, some wildly curved aspheric surfaces will not converge if this "fast guess" is used. For systems which use these surfaces, this box may need to be unchecked, in which case ZEMAX will use a slower but more robust algorithm for finding the solution. Whether the box is checked or unchecked, ZEMAX will find an accurate ray intercept solution, or an error message will be issued.

### 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" on page 51) 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 semi-diameter 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"

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. 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.

### Semi Diameter Margin (lens units)

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.

The semi diameter margin control allows specification of an additional amount of radial aperture as a fixed number in lens units. The default value of zero leaves no margin, while a margin of 2.0 lens units would add 2.0 lens units to the semi-diameters of all surfaces under "automatic" control.

See "Semi Diameter Margin %" below. If both "percent" and "lens units" margin values are non-zero, the percent is added first, then the lens unit margin.

### 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.

The 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%.

See "Semi Diameter Margin (lens units)" above. If both "percent" and "lens units" 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 conversion can be written 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 global coordinate, "o" indicates an offset coordinate (the translation) and "l" indicates local coordinate. The rotation matrix R and the offset vector can be computed for any surface using any other surface as a global reference.

The rotation matrix yields valuable insight into the orientation of the surface coordinate system with respect to the global reference surface. On the local surface, the unit vector which is oriented along the local x direction is

(1, 0, 0 ). This vector may be rotated using the R matrix to yield the orientation of the local x axis in the global coordinate system. A matrix multiplication for each of the three local axis unit vectors yields

$$\hat{x}_l = \begin{bmatrix} R_{11} \\ R_{21} \\ R_{31} \end{bmatrix}, \hat{y}_l = \begin{bmatrix} R_{12} \\ R_{22} \\ R_{32} \end{bmatrix}, \hat{z}_l = \begin{bmatrix} R_{13} \\ R_{23} \\ R_{33} \end{bmatrix}.$$

Note that the global coordinate system orientation of the unit vectors oriented along the local coordinate axes are the columns of the R matrix. The global vertex listing on the prescription report lists the R matrix components and offset vector for each surface, all referenced to the global reference surface. If the surface is a coordinate break, then the R matrix includes the effects of the coordinate translation and rotation. If the reference surface is a coordinate break, the reference coordinate system is defined after the decentration and rotation of the local coordinate system. If the surface for which R is computed precedes the reference surface, and it is a coordinate break, the R matrix is for the system prior to the coordinate transformation. When in doubt, insert a dummy surface that is not a coordinate break at the location of interest to check the global orientation.

The reference surface will be incremented or decremented automatically as surfaces above the reference surface are inserted or deleted. Any surface may be chosen as the reference surface, unless the object is at infinity, then 0 may not be the reference surface. The reference surface is also used to define the point of overlap for multiple zoom positions on the 3D layout plot.

### Method To Compute F/#

By default, ZEMAX computes the paraxial and working F/# of a system using ray tracing. For details, see "Paraxial working F/#" on page 51 and "Working F/#" on page 55. 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 effect the calculation of F/#. For these reasons, ZEMAX will "cap" the maximum F/# to a value of 10,000. It is strongly recommended that rays be used to compute the F/# for all systems whose F/# is less than 10,000.

The preferred method of modeling systems with very large F/#s is to use a paraxial lens in the optical space preceding the image surface with any convenient focal length selected to "convert" the nearly collimated beam to a focusing beam with a more moderate F/#. This method avoids round-off error problems associated with nearly collimated rays traveling great distances. The only disadvantage to this approach is that the effective magnification of the paraxial lens must separately be accounted for. For example, if the true F/# of the system is 20,000, a paraxial lens may be selected to convert the beam to a F/20 system. However, the diffraction results ZEMAX computes based upon the F/#, such as the Point Spread Function width, need to be interpreted as being scaled by the additional factor of 1000.

Optionally, the F/# can be computed using the exit pupil distance divided by the exit pupil diameter. This method can be selected using this "Method To Compute F/#" switch. 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. Therefore, this option should only be used when the chief ray at all field position makes a small angle with respect to the Z axis in image space. Second, the exit pupil diameter and position used are based upon the paraxial values for these parameters (see "Exit pupil diameter" on page 47). 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 "Ray Aiming" on page 88) and float by stop size for the aperture type (see "Aperture Type" on page 85).

### Non-Sequential

The controls on this tab 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.

### 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.

### 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. This can affect ray tracing results when tracing rays from faceted reflectors, and a ray strikes within the glue distance of the edge between two angled facets.

In the majority of cases, no adjustment should be made to the glue distance parameter. The glue distance must be no smaller than 1E-10 and no larger than 1.0.

### Missed Ray Draw Distance in Lens Units

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.

### 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.

## **Fields**

The field dialog box allows specification of field points. Field points may be specified as angles, object heights (for systems with finite conjugates), paraxial image heights, or real image heights. The buttons can be used to activate or deactivate field positions, as well as to sort, save and load the data. For information about field conventions, see "Field angles and heights" on page 47.

### Vignetting factors

ZEMAX also provides data fields for defining vignetting factors. The vignetting factors are VDX, VDY, VCX, VCY, and VAN. These vignetting factors should be left at zero if there is no vignetting in the system. The factors are described in the Chapter "Conventions and Definitions" under "Vignetting factors" on page 53.

There is also a button labeled "Set Vig" on the fields dialog box. Clicking on this button will recompute the vignetting factors for each field based upon the current lens data. 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. To clear the vignetting factors back to the default values of zero, click on "Clr Vig". The set vignetting algorithm may not work properly if there are central or non-circular obscurations or apertures in the system. In this case, the vignetting factors must be user defined.

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.



***Identical or nearly identical field coordinates may not be defined with different vignetting factors; see the Chapter Conventions and Definitions for details.***

---

### Saving and loading field data

The Save and Load buttons on the field dialog box 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.

## **Wavelengths**

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" button.



For more information, see the “Wavelength data” on page 55.

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.

### **Saving and loading wavelength data**

The Save and Load buttons on the wavelength data dialog box 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.

### **Next Configuration**

This menu option provides a quick way to change all graphics to reflect the next configuration (or zoom position). All spreadsheet, text, and graphics data will be updated.

### **Last Configuration**

This menu option provides a quick way to change all graphics to reflect the last configuration (or zoom position). All spreadsheet, text, and graphics data will be updated.



## **Introduction**

This chapter provides detailed descriptions of each of the analysis features ZEMAX supports. Analysis in this context means any graphical or text data computed from data defining the lens. This includes aberrations, MTF, spot diagrams, and many other computations. Program features which modify the lens data or which manipulate other data (such as glass catalog data) are described in the chapter "Tools Menu".

Selecting a menu option will immediately perform the requested calculation. Once the graph or text window is displayed, you may select the Settings menu option to modify the default settings for that window. Once you have made the appropriate changes, click on "OK" and the program will recalculate and redisplay the data presented in the window. If you prefer to change the settings before the graphic or text data is displayed, use the "Show Options First" checkbox on the Graphics tab of the File, Preferences dialog box.

For a description of the OK, Cancel, Save, Load, Reset, and Help buttons present on most of the "Settings" dialog windows, see the chapter "User Interface".

Each analysis window has an "Update" menu item. The update function forces ZEMAX to recompute and redisplay the data presented in the window. This is useful if the lens data has changed and the graph now displays obsolete data. Double clicking within the window has the same effect as selecting Update. Clicking with the right mouse button is equivalent to clicking on "Settings". For more information, see "USER INTERFACE" on page 35.

## **Layout**

### **2D Layout**

#### *Purpose:*

Layout diagram. This is a YZ cross section through the lens.

#### *Settings:*

Item	Description
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 screen). 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.

Item	Description
Color Rays By	Select "Fields" to use color to distinguish between each field position, or "Waves" to distinguish between each wavelength.
Suppress Frame	Suppresses drawing of the frame on the bottom of the screen, 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.

### *Discussion:*

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" on page 174.

## 3D Layout

### *Purpose:*

Draws 3D layout plots of the lens system. The algorithm draws a wireframe style representation of the lens.

### *Settings:*

Item	Description
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 screen). 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.

Item	Description
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, "Fields" will use color to distinguish between each field position, "Waves" to distinguish between each wavelength, or "Config" to distinguish between configurations. For non-sequential rays "Fields" 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 screen, 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 draw all configurations at once, or select the one configuration to draw, or select "Current" to show the active configuration.
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.
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 (see "Global Coordinate Reference Surface" on page 93). This 3D model is then rotated by the Rotation About X, Y, and Z values specified in the settings (see also "Graphics" on page 59 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" on page 174. 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 on the Miscellaneous tab of the System, General dialog box. 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 main ZEMAX directory. 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 px and one for the py 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

### Wireframe

#### *Purpose:*

Draws a representation of the lens.

#### *Settings:*

The options are similar to those available for the 3D layout feature. The "Hide Lens Edges" and "Hide X Bars" checkboxes are not available, and several new controls are added as described below.

Item	Description
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.
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.

#### *Discussion:*

The wireframe algorithm describes the lens as a collection of polygon facets. The number of facets used to display the lens elements can be modified using the radial and angular segment options. The wireframe model

is identical to the solid model, except hidden lines are not removed. This representation may cause the screen to become cluttered with lines. The "Hide Lens Faces" option can be used to clean up the display. The advantage to this display method is speed; it is faster than the solid model.

Pressing the left, right, up, down, Page Up, or Page Down keys will rotate the displayed image for a different perspective.

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, see "Ray Trace" on page 174.

See also "The orientation indicator" on page 101.

## **Solid Model**

### *Purpose:*

Draws a hidden-line representation of the lens.

### *Settings:*

The options are similar to those available for the 3D layout feature. The "Hide Lens Edges" and "Hide X Bars" checkboxes are not available, and several new controls are added as described below.

Item	Description
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.
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.

### *Discussion:*

The solid model algorithm describes the lens as a collection of polygon facets. The lines and facets which are hidden from view are removed, which gives the lens a solid appearance. This algorithm is slower than the other layout plots. The number of facets used to display the lens elements can be modified using the radial and angular segment options.

Pressing the left, right, up, down, Page Up, or Page Down keys will rotate the displayed image for a different perspective.

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, see "Ray Trace" on page 174.

See also "The orientation indicator" on page 101.

## **Shaded Model**

### *Purpose:*

Draws a shaded solid model representation of the lens using OpenGL graphics.

### *Settings:*

The options are almost identical to those available for the Solid Model feature, except there are additional controls for setting the lighting level and background color.

There is an additional control labeled "opacity". The options are to ignore, which ignores the surface and object opacity values and draws all surfaces and objects as opaque; and method 1 or 2, which use different algorithms to render the scene. Some systems are difficult for OpenGL to render when there are a mixture of opaque and transparent surfaces and objects; choose the method which yields the preferred rendering. For information on setting the opacity of surfaces and objects see "Surface Opacity" on page 67 and "The object properties dialog box" on page 348.

## ZEMAX Element Drawing

### *Purpose:*

This feature creates a mechanical drawing of surface, singlet, cemented doublet, or cemented triplet elements suitable for use in optical shop fabrication.

### *Settings:*

Item	Description
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 factor of 1.0 will plot the element at actual size on the printer (not the screen). 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 ASCII 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.

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

$$\#fringes = \frac{\Delta R \rho^2}{\lambda R^2},$$

where  $\Delta R$  is the radius error,  $\lambda$  is the test wavelength,  $\rho$  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

#### *Purpose:*

This feature creates an ISO 10110 type drawing of surface, singlet, or doublet elements suitable for use in optical shop fabrication.

#### *Settings:*

Item	Description
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.

#### *Discussion:*

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.

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.

### 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 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))

Symbol or Code Number	Description
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 <sup>2</sup> .

### 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, TTHI, TIND, TABB, TSTX, TSTY, TSDX, and TSDY 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.

### NSC 3D Layout

#### *Purpose:*

Draws 3D layout plots of the sources and objects in a single NSC group.

#### *Settings:*

Item	Description
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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details.
Suppress Frame	Suppresses drawing of the frame on the bottom of the screen, which leaves more room for the layout plot itself. No scale bar, address block, or other data will be displayed.
Configuration	Select "All" to draw all configurations at once, or select the one configuration to draw, or select "Current" to show the active configuration.
Color Rays By	Select "Sources" to use color to distinguish rays traced from each source, unless a source has a user defined color assigned to that source, or "Waves" to distinguish between each wavelength, or "Config" to distinguish between configurations.
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 screen). A factor of 0.5 will plot the lens at half scale.

Item	Description
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.
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" on page 371 for information about the filter string syntax.
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 files" on page 366.

#### *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" on page 67.

### **NSC Shaded Model**

#### *Purpose:*

Draws a shaded solid model representation of the lens using OpenGL graphics.

#### *Settings:*

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" on page 332 for details.

There is an additional control labeled "opacity". The options are to ignore, which ignores the surface and object opacity values and draws all surfaces and objects as opaque; and method 1 or 2, which use different algorithms to render the scene. Some systems are difficult for OpenGL to render when there are a mixture of opaque and transparent surfaces and objects; choose the method which yields the preferred rendering. For information on setting the opacity of surfaces and objects see "Surface Opacity" on page 67 and "The object properties dialog box" on page 348.

### **Fans**

#### **Ray Aberration**

#### *Purpose:*

Shows ray aberrations as a function of pupil coordinate.

## Settings:

Item	Description
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" on page 109.

## 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 plane 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 is not appropriate, including those systems which require significant use of 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, and changes the system aperture to an entrance pupil diameter equal to the original paraxial entrance pupil diameter computed for the original stop position. If the selected surface follows the stop surface, no changes are made to the stop definition.

Then, surfaces which follow the selected surface are 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 intermediate surfaces 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.

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

### *Purpose:*

Shows optical path difference as a function of pupil coordinate.

### *Settings:*

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

### *Purpose:*

Shows entrance pupil distortion as a function of pupil coordinate.

### *Settings:*

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" on page 88) 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.

## **Spot Diagrams**

### **Standard**

#### *Purpose:*

Show spot diagrams.

#### *Settings:*

Item	Description
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.
Show Scale	Scale bar is the default. Selecting "Airy Disk" will draw an elliptical ring around each spot showing the size of the Airy ellipse. The Airy disk radius is 1.22 times the wavelength (primary wavelength is used if polychromatic) times the F/# of the system; which in general depends upon field position and pupil orientation. If the Airy disk is larger than the spot, the Airy disk will set the scale size, otherwise the spot radius will set the scale. 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 an appropriate scale.
Delta Focus	The delta focus option is only used if through-focus spot diagrams are selected. This is the Z-axis spacing between spot diagram planes. Five spot diagrams will be shown for each field angle. The defocus will be -2, -1, 0, 1, and 2 times the delta focus provided. The delta focus units are in micrometers. For some systems the default delta focus may be too small to cause a change in the spot diagram structure.
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.

Item	Description
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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Scatter Rays	If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. Only ZEMAX-EE supports this capability.
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 draw all configurations at once, or select the one configuration to draw, or select "Current" to show the active configuration.
Color Rays By	Select "Fields" to use color to distinguish between each field position, or "Waves" to distinguish between each wavelength, or "Config" to distinguish between configurations.

### *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 vignettted 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 plane intercept coordinates of the reference point are shown underneath each spot diagram. If a surface other than the image plane 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.

### Through Focus

#### *Purpose:*

Show spot diagrams as they change through focal plane shifts.

#### *Settings:*

The options are identical to the standard spot diagram.

#### *Discussion:*

The through focus spot diagrams are useful for estimating astigmatism, or for analyzing best focus or depth of focus.

### Full Field

#### *Purpose:*

Shows spot diagram with all field points on a common scale.

#### *Settings:*

The options are identical to the standard spot diagram.

#### *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 useless if the spot radius is small compared to the total field size, because in this case the spots for each field will appear as "dots". If "chief ray" is selected as the reference point, then the chief ray for field position 1 will be used.

### Matrix

#### *Purpose:*

Show spot diagram as a matrix of individual diagrams, with each field along a row and each wavelength down a column.

#### *Settings:*

The options for the matrix spot diagram are similar to those for the standard spot diagram, except for the addition of the following option.

Item	Description
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

#### *Purpose:*

Show spot diagram as a matrix of individual diagrams, with each field along a row and each configuration down a column.

#### *Settings:*

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.

## **MTF**

### **FFT MTF**

#### *Purpose:*

Computes the diffraction modulation transfer function (MTF) data for all field positions using an FFT algorithm.

#### *Settings:*

Item	Description
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 spatial frequency (in cycles per millimeter) 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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Use Dashes	Selects either solid lines or dashed lines to differentiate the various curves.
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" on page 109.

#### *Discussion:*



***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:

$$S(v) = \frac{4}{\pi} \left[ \frac{M(v)}{1} - \frac{M(3v)}{3} + \frac{M(5v)}{5} - \frac{M(7v)}{7} + \dots \right]$$

where  $S(v)$  is the square wave response,  $M(v)$  is the sinusoidal modulation response, and  $v$  is the spatial frequency.

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.

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” on page 118.

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 off 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” on page 46.

The spatial frequency scale of the MTF plot is always in cycles per mm in image space, which is the correct term for sinusoidal MTF response. The term line pairs per mm is often used, but strictly speaking line pairs per mm only applies to bar, as opposed to sinusoidal targets. ZEMAX uses these terms interchangeably, as is common in the industry. MTF is always measured 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” on page 119 and the “Huygens MTF” on page 118.

## FFT Through Focus MTF

### *Purpose:*

Computes the FFT modulation transfer function data as a function of focus shift at a specific spatial frequency.

### *Settings:*

Item	Description
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	The range of defocus used.
Frequency	The spatial frequency (in cycles per millimeter) 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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Use Dashes	Selects either solid lines or dashed lines to differentiate the various curves.

### *Discussion:*

See “FFT MTF” on page 114 for details.

## **FFT Surface MTF**

### *Purpose:*

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.

### *Settings:*

Item	Description
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.
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.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

### *Discussion:*

The regular MTF plot is just two orthogonal cross sections through the surface MTF plot. This plot is primarily qualitative. See “FFT MTF” on page 114 for details.

## **FFT MTF vs. Field**

### *Purpose:*

Computes the FFT MTF data as a function of field position, and displays the data in a graph.

### *Settings:*

Item	Description
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 (in cycles per millimeter) for which data is plotted.
Wavelength	The wavelength number to be used in the calculation.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Use Dashes	Selects either solid lines or dashed lines to differentiate the various curves.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See the comments about vignetting factors in the discussion section below.

Item	Description
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” on page 114 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” on page 96 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**

**Purpose:**

Computes the FFT MTF as a function of field position, and displays the data over a rectangular region of field.

**Settings:**

Item	Description
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 at which to compute the MTF.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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.

Item	Description
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” on page 117.

*Discussion:*

See “FFT MTF” on page 114 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 feature on page 121.

## Huygens MTF

*Purpose:*

Computes the diffraction modulation transfer function (MTF) data using a Huygens direct integration algorithm.

*Settings:*

Item	Description
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” on page 126.
Image Delta	The distance in micrometers between points in the image grid.
Zero Padding	Adds a zero-value guard band around the computed PSF before performing the Fourier transform.
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” on page 471. This coherent sum also assumes that the image surface is located in the identical position in all configurations.
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 spatial frequency in cycles per millimeter to display.
Show As	Choose surface plot, contour map, grey scale, or false color map as the display option.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Use Dashes	Selects either solid lines or dashed lines to differentiate the various curves.

*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 initial PSF sampling settings for Image Sampling and Image Delta settings are the same as for the Huygens PSF, therefore it is instructive to do a Huygens PSF first (see “Huygens PSF” on page 126). The zero padding setting adds zero intensity values around the PSF and has the visual effect of increasing the pixel density in the transform. 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 spatial frequency scale of the MTF is always in cycles per mm in image space.

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” on page 114.

### **Huygens Through Focus MTF**

#### *Purpose:*

Computes the diffraction modulation transfer function (MTF) data using a Huygens direct integration algorithm and displays the data as a function of delta focus.

#### *Settings:*

Item	Description
Delta Focus	The range of defocus used.
Frequency	The spatial frequency (in cycles per millimeter) 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.

The other settings are the same as for the Huygens MTF feature, as described on page 118.

#### *Discussion:*

This feature is very similar to the Huygens MTF feature, as described on page 118.

### **Huygens Surface MTF**

#### *Purpose:*

Computes the diffraction modulation transfer function (MTF) data using a Huygens direct integration algorithm and displays the data as a surface, grey scale, false color, or contour plot.

#### *Discussion:*

This feature is very similar to the Huygens MTF feature, as described on page 118.

### **Geometric MTF**

#### *Purpose:*

Computes the geometric MTF, which is an approximation to the diffraction MTF based upon ray aberration data.

### Settings:

Item	Description
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 (in cycles per millimeter) 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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Scatter Rays	If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. Only ZEMAX-EE supports this capability.

### 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" on page 114.

## Geometric Through Focus MTF

### Purpose:

Computes the geometric MTF data through focus at a specific spatial frequency.

### Settings:

Item	Description
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	The range of defocus used.
Frequency	The spatial frequency (in cycles per millimeter) 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.



Item	Description
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Scatter Rays	If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. Only ZEMAX-EE supports this capability.
Use Dashes	Selects either solid lines or dashed lines to differentiate the various curves.

*Discussion:*

See the Geometric Transfer Function section on page 119 for details.

### **Geometric MTF vs. Field**

*Purpose:*

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.

*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**

*Purpose:*

Computes the geometric modulation transfer function data as a function of field position, and displays the data over a rectangular region of field.

*Settings:*

Item	Description
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 at which to compute the GMTF.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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.

Item	Description
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 ( $h_x = 0$ , $h_y = 0$ ), 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. Only ZEMAX-EE supports this capability.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See “Comment about vignetting factors” on page 117.

#### *Discussion:*

See the Geometric Transfer Function on page 119 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 (Diffraction) MTF Map feature on page 117.

## **PSF**

### **FFT PSF**

#### *Purpose:*

Computes the diffraction point spread function (PSF) using the Fast Fourier Transform (FFT) method.

#### *Settings:*

Item	Description
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), or phase.
Show As	Choose surface plot, contour map, grey scale, or false color map as the display option.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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” on page 109.

### *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" on page 126.

### 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 planes, 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 plane 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 plane 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/#),  $\lambda$  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.

The image delta,  $\Delta$ , 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" on page 126.

## FFT PSF Cross Section

### *Purpose:*

This feature plots cross sections through the diffraction PSF.

### *Settings:*

Item	Description
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, or phase. X cross sections are called rows, and Y cross sections are called columns, but this is arbitrary
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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:*



***See the discussion section of the FFT PSF. Those comments also apply to this feature.***

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.

## FFT Line/Edge Spread

### *Purpose:*

This feature plots the edge or line spread functions based upon a computation and integration of the diffraction FFT PSF.

### Settings:

Item	Description
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 or logarithmic scale.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

### Discussion:



**See the discussion section of the FFT PSF. Those comments also apply to this feature.**

The Line Spread Function is computed by integration of the PSF along either rows (for Y orientation) or columns (for X orientation). The Edge Spread Function is the subsequent running integral of the Line Spread Function. The X orientation refers to the line or edge being oriented parallel to the X axis of the PSF. The Y orientation refers to the line or edge being oriented parallel to the Y axis of the PSF. See also "Geometric Line/Edge Spread" on page 137.

## Huygens PSF

### Purpose:

Computes the diffraction PSF using direct integration of Huygens wavelets method. The Strehl ratio is also computed.

### Settings:

Item	Description
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.
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).

Item	Description
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" on page 471. This coherent sum also assumes that the image surface is located in the identical position in all configurations.
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.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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. 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 plane 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.

### *Huygens PSF Cross Section*

#### *Purpose:*

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" on page 126, with the difference being the data is plotted as a cross section.

## **Wavefront**

### **Wavefront Map**

#### ***Purpose:***

Displays the wavefront aberration.

#### ***Settings:***

Item	Description
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.
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	By default, the shape of the pupil is distorted to show the approximate shape of the exit pupil as seen from the on axis chief ray image point. 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" on page 109.
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.

#### ***Discussion:***

See also the Interferogram feature described below.

### **Interferogram**

#### ***Purpose:***

Generates and displays interferograms.

#### ***Settings:***

Item	Description
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" on page 109.
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).



Item	Description
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.

#### *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**

#### *Purpose:*

Generates and displays Foucault knife-edge shadowgrams.

#### *Settings:*

Item	Description
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.

Item	Description
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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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 any position near focus; 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 BMP or JPG 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.

## **Surface**

### **Surface Sag**

#### *Purpose:*

Displays the sag of a surface as a 2D color or contour map, or as a 3D surface plot.

### Settings:

Item	Description
Sampling	The size of the grid used. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase. ZEMAX actually increases the sampling by 1, so that an odd sampling number is used. This allows for a more symmetric display of the data.
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.
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

#### Purpose:

Displays the phase of a surface as a 2D color or contour map, or as a 3D surface plot.

### Settings:

Item	Description
Sampling	The size of the grid used. The sampling may be 32x32, 64x64, etc. Although higher sampling yields more accurate data, calculation times increase. ZEMAX actually increases the sampling by 1, so that an odd sampling number is used. This allows for a more symmetric display of the data.
Contour Format	The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String" on page 131. The contours are defined in periods; each period is a phase change of $2\pi$ .
Surface	The surface number to compute the phase display for.
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  $2\pi$ . 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.

## **RMS**

### **RMS vs. Field**

#### ***Purpose:***

Plots RMS radial, x, and y spot radius, RMS wavefront error, or Strehl ratio as a function of field angle.

#### ***Settings:***

Item	Description
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	Selects either Gaussian quadrature or rectangular array. The Gaussian quadrature method is very fast and accurate, but only works if there is no vignetting. If any rays are to be vignetted, then rectangular array is more accurate.
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 167. For a polychromatic Strehl, see the "Huygens PSF" on page 126.
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, select any one wavelength to plot monochromatic data, or select "Poly Only" which only plots 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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See "Comment about vignetting factors" on page 117.

#### ***Discussion:***

This feature calculates the RMS error or Strehl ratio as a function of field angle for each wavelength.

Two different methods of calculation are used; either a Gaussian quadrature method, or a rectangular array of rays. For the Gaussian quadrature method, 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 rectangular array method, and a larger number of rays for sufficient accuracy.

### Comments about RMS wavefront computations

For RMS wavefront computations, ZEMAX always subtracts out the mean OPD. RMS wavefront, when referenced to the centroid, also requires computation of a shifted, titled 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

#### *Purpose:*

Plots RMS radial, x, and y spot radius, RMS wavefront error, or Strehl ratio as a function of wavelength.

#### *Settings:*

Item	Description
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	Selects either Gaussian quadrature or rectangular array. The Gaussian quadrature method is very fast and accurate, but only works if there is no vignetting. If any rays are to be vignetted, then rectangular array is more accurate.
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.

Item	Description
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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

#### *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" on page 132; 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**

#### *Purpose:*

Plots RMS radial, x, and y spot radius, RMS wavefront error, or Strehl ratio as a function of focus change.

#### *Settings:*

Item	Description
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	Selects either Gaussian quadrature or rectangular array. The Gaussian quadrature method is very fast and accurate, but only works if there is no vignetting. If any rays are to be vignetted, then rectangular array is more accurate.
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" on page 167. For a polychromatic Strehl, see the "Huygens PSF" on page 126.
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. The units are lens units.

Item	Description
Max Focus	The maximum value of the defocus to plot. The units are lens units.
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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

#### *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" on page 132; 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 plane. 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 plane farther away from the last component. For systems with an even number of mirrors, then negative focus values move the image plane closer to the last component.

## **Encircled energy**

### **Diffraction**

#### *Purpose:*

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.

#### *Settings:*

Item	Description
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" on page 109.
Wavelength	The wavelength number to be used in the calculation.
Field	The field number for which the calculation should be performed.

Item	Description
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.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

#### 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”).

### Geometric

#### Purpose:

Computes encircled energy using ray-image plane intercepts.

#### Settings:

Item	Description
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” on page 109.
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.



Item	Description
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Scatter Rays	If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. Only ZEMAX-EE supports this capability.

*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

*Purpose:*

Computes the geometric response to a line object and an edge object.

*Settings:*

Item	Description
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” on page 109.
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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

*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 tangential and sagittal data refer to the orientation of the line or edge. The zero coordinate is referenced to the chief ray. This is a geometric calculation. See also “FFT Line/Edge Spread” on page 125.

## Extended Source

*Purpose:*

Computes encircled energy using an extended source similar to the geometric image analysis feature.

## Settings:

Item	Description
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	<p>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.</p> <p>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.</p>
Refer To	Select either chief ray or centroid 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" on page 109.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Multiply by 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.
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 \ImaFiles directory. 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.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See "Comment about vignetting factors" on page 117.

## 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.

## **Illumination**

### **Relative Illumination**

#### ***Purpose:***

Computes the relative illumination as a function of radial field coordinate for a uniform Lambertian scene. This feature also computes the Effective F/#.

#### ***Settings:***

Item	Description
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 $10 \times 10 \times \pi / 4$ 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.
Wavelength	Selects the wavelength for computation. Relative illumination is a monochromatic entity.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Log Scale	If checked, a logarithmic rather than linear scale will be shown.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See "Comment about vignetting factors" on page 117.
Scan Type	Choose +y, +x, -y, or -x field scan direction.

#### ***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 pupils, 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.

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 law curve, because the cosine-fourth law is an approximation based upon a thin, aberration free lens with the stop at the lens illuminating a plane image surface. For general systems including telecentric systems, systems with pupil or image aberrations, or vignetting,

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.

For systems with very high amounts of vignetting, or for systems with non-linear cosine space aberrations that would violate the assumptions of the computation, the relative illumination cannot be calculated, and an error message will be displayed. Cosine space aberrations may be displayed using the spot diagram feature.

### 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.

### Vignetting Plot

#### *Purpose:*

Calculates fractional vignetting as a function of field angle.

#### *Settings:*

Item	Description
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" on page 117.

#### *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 plane, 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 vignettted.

See also the relative illumination feature.

### Illumination XY Scan

#### *Purpose:*

Computes relative illumination for an extended source along a line across any surface.

## Settings:

Item	Description
Sampling	Selects the grid size used to sum the illumination. The sampling determines how many pixels are used to collect the ray data.
Rays x 1000	Determines the approximate total number of rays to be traced in computing the illumination due to the extended source.
File	The name of the IMA file used to define the shape of the extended source. See the Geometric Image Analysis feature for details.
Source Size	The full width in field units of the extended source.
Rotation	The angle to rotate the extended source in object space about the normal to the center of the extended source.
Show As	Select either X or Y direction scan.
Smoothing	Smoothing helps remove jagged lines due to low ray sampling by applying a smoothing operator to average the data in adjacent pixels.
Wavelength	The wavelength number to be used in the calculation.
Field	The field number indicates which field position is used as the center reference point for the extended source.
Surface	The scan may be computed at any surface, however, the relative illumination computation is only used at the image surface.
Detector Size	The total width of the detector in lens units. The detector size is divided into pixels according to the "Sampling" setting.
Use Relative Illumination	If checked, then the RI computation described in the feature "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. RI may only be used if the selected surface is the image surface, and ZEMAX assumes the image surface is a reasonable conjugate of the object surface. See the RI feature for details. If unchecked, the RI is estimated from the system distortion. RI is only considered if the surface is the image surface.
Assume Lambertian Source	If checked, rays are weighted according to the cosine of the angle they make with respect to the object surface. This cosine factor is not appropriate in all cases. This option is not available if "Use Relative Illumination" is checked, since the RI calculation already assumes the object is Lambertian.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See "Comment about vignetting factors" on page 117.

## Discussion:

The illumination XY scan is similar to the relative illumination (RI) feature, with the added capability to estimate the RI for non-uniform extended sources. For uniform extended Lambertian sources, the RI feature is faster and more accurate. However, for systems with complex source properties, the illumination XY scan can estimate illumination by Monte Carlo ray tracing combined with the conventional RI computation.

The extended sources are defined in the same way as described in the Image Analysis feature.

## **Illumination 2D Surface**

### **Purpose:**

Computes relative illumination for an extended source over a 2D surface.

### **Settings:**

The options are identical to the XY scan, except that a 2D surface is drawn as either an isometric surface or contour plot, or as a grey scale or false color map.

### **Discussion:**

See the illumination XY scan feature.

## **Image Analysis**

### **Geometric Image Analysis**

### **Purpose:**

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.

This feature is based strictly upon geometrical ray tracing; see the Diffraction Image Analysis for a diffraction based feature with similar functionality.

The ability of this feature to render color images is limited, for analyzing color bitmap images see "Geometric Bitmap Image Analysis" on page 146.

The image analysis feature uses special IMA or BIM files to describe the object to be imaged. The IMA and BIM file formats are described in the discussion section which follows.

### **Settings:**

Item	Description
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, or spot diagram as the display option.

Item	Description
Source	The source may be uniform or Lambertian. The uniform setting weights all rays equally. Lambertian weights all rays by the cosine of the angle the ray makes with the axis of the object surface.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
Scatter Rays	If checked, rays will be statistically scattered at ray-surface intercepts that have defined scattering properties. Only ZEMAX-EE supports this capability.
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 \ImaFiles directory. 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.
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 draw all configurations at once, or select the one configuration to draw, or select "Current" to show the active configuration.
Reference	Selects the reference coordinate for the center of the plot, either the chief ray or the surface vertex.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See "Comment about vignetting factors" on page 117.
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 \ImaFiles subdirectory.

#### *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 ASCII 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 ASCII 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  $n \times n$  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 ASCII 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,  $n_x$ .

1 32 bit integer representing the number of y pixels,  $n_y$ .

followed by  $n \times n$  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  $n_x$  and  $n_y$  values must be identical or an error message will be issued.

### 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 ASCII 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 provided on the wavelength data screen. 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.



The field size determines the physical size of the image file as seen by the optical system. For example, if a 30 x 30 pixel size image file is used, and the field size is 2.0 mm (this assumes the fields are defined in terms of object or image heights), then each pixel represents a 66.67 micrometer region. If the same image file is later used with a system with a 40 degree full field of view (using field angles), then the field size 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" on page 47.

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.

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 vignettted 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" on page 174. Also see "Computing Fiber Coupling" on page 530.

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 an ASCII 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

#### *Purpose:*

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 Windows BMP and JPG files as the source image, see the discussion for details.

#### *Settings:*

Item	Description
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 x 1000	This setting determines how many rays will be traced. The distribution of rays at each wavelength is in proportion to the relative RGB intensities of each pixel.
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.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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.

Item	Description
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 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 wavelength, such as 1, 2, 3, etc... the B, G, or R channel image will be used; for wavelengths higher than 3 the B channel is always used, for wavelengths selected which are not defined the highest defined wavelength will be 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 or .JPG source image file. This file must reside in the \lmaFiles directory.
Surface	The surface number at which to evaluate the rays.
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.
Show	Select either "object" or "image". If object is selected, then the source bitmap will be drawn. If image is selected then rays will be traced and the detected image will be displayed. Note the number of rays, pixels, and pixel sizes are ignored when drawing the object bitmap.
Output	The name of the .BMP or .JPG 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 either a BMP or JPG extension, with no path name supplied. This file will be created or overwritten without warning and will be placed in the \lmaFiles directory.
Remove Vignetting Factors	If checked, vignetting factors are automatically removed. See "Comment about vignetting factors" on page 117.

### Discussion:

See also the Geometric Image Analysis feature discussion, as this feature is very similar.

The BMP files used as source bitmaps must be standard Windows format, uncompressed, 24-bit color RGB bitmaps. Not all files ending in the .BMP extension meet this definition. Contact technical support if you have a BMP file that does not work correctly for advice on converting the file to the proper format. Any properly formatted JPG file will work.

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.

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.

The pixels from which rays are traced are chosen randomly according to the RGB intensity of each pixel in each color channel. Rays are generated randomly from anywhere within the bitmap until the specified number of rays have been launched, although not all these rays may make it through the lens.

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.

Selecting the "Text" option on the menu bar will generate and display an ASCII file listing the detected pixel data.

Use the "Escape" key to terminate a lengthy image analysis computation.

## Diffraction Image Analysis

### *Purpose:*

The diffraction image analysis feature is similar to the geometric image analysis feature, except the complex system Optical Transfer Function (OTF) is used 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.

The image analysis feature uses IMA/BIM files to describe the object to be imaged. See "The IMA format" on page 144 and "The BIM format" on page 144.

### *Settings:*

Item	Description
File Size	The full width in lens units of the region defined by the IMA file measured in image space. Note IMA files are always square.
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.
Zero Padding	Determines the actual size of the region to compute the diffraction image of by adding zero intensity values around the IMA 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 IMA 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 IMA file has an even number of pixels, the modified image will have $2*n$ pixels.
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, or false color map as the display option.

Item	Description
Data Type	Choose either the incoherent image, coherent image, raw image, incoherent transfer function, coherent transfer function, or the transform of the raw image. See the discussion below for information about the coherent transfer function and the limitations of this computation.
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 image file. This file must reside in the \ImaFiles directory. See the discussion section for a full description of the IMA file format.
Edit IMA File	Pressing this button will invoke the Windows Notepad editor which allows modification of the currently selected IMA file.
Contour Format	The contour format string. For a discussion of the contour format string syntax, see "The Contour Format String" on page 131. The contours are defined in units of relative intensity.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

#### 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 IMA file defines the relative intensity at each of an arbitrary number of pixels arranged on a square grid. For example, the letter "F" may be described by a 7 x 7 grid such as:

```
0111110
0100000
0100000
0111100
0100000
0100000
0100000
```

The file size parameter determines how big each pixel 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, which may be either object or image space. For the diffraction image 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.

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
00110000000000
00111111110000
00111111110000
00110000000000
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:

```

00000000000000
00000000000000
00000000000000
0000111110000
0000100000000
0000100000000
0000111100000
0000100000000
0000100000000
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.

Once the input image is defined, the image is transformed into frequency space, multiplied by the OTF, and transformed back into position space. The resulting image is filtered by the complex OTF to yield the diffraction image. 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 "Extended Diffraction Image Analysis" on page 151). 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 "Geometric Image Analysis" on page 142 or "Extended Diffraction Image Analysis" on page 151.

Note that the diffraction image analysis feature is good at computing detailed image data for small images, while the geometric image analysis feature is good at computing image data for larger scale images.

### 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.

### Extended Diffraction Image Analysis

#### *Purpose:*

The extended diffraction image analysis feature is similar to the diffraction image analysis feature, except the Optical Transfer Function (OTF) may vary over the field of view of the image. This feature uses IMA/BIM files to describe the object to be imaged. See “The IMA format” on page 144 and “The BIM format” on page 144.

#### *Settings:*

Item	Description
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 and the OTF sampling (see the discussion below).
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	If smaller than the maximum image size available (see the discussion below) then this parameter will define 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” on page 151 for information about the coherent transfer function and the limitations of this computation.
Diffraction Limited	If checked, aberrations are ignored. Apertures are still considered.

Item	Description
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.
File	The name of the image file. This file must reside in the \ImaFiles directory. See "The IMA format" on page 144 and "The BIM format" on page 144 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" on page 131. The contours are defined in units of relative intensity.
Use Polarization	If checked, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the "System Menu" chapter under "Polarization" for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.
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\BeamFiles subdirectory. See "ZEMAX Beam File (ZBF) binary format" on page 528.

### 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 the Diffraction Image Analysis feature applies, see "Diffraction Image Analysis" on page 148 for details. The differences between the Diffraction Image Analysis feature and the Extended Diffraction Image Analysis feature 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. Convoluting this function with the system PSF (see "FFT PSF" on page 122) 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 Diffraction 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  $n \times n$ , ZEMAX will actually store  $n \times n + 1$  OTF's: the  $n \times n$  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  $64 \times 64$ , a single OTF will be  $128 \times 128$ , using up 256k of RAM ( $128 \times 128 \times 2 \times 8$  bytes). For an OTF grid of  $5 \times 5$ , the total comes to  $26 \times 256k = 6.5$  Mb. For  $128 \times 128$  sampling and a  $9 \times 9$  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, which may be either object or image space. For the diffraction image 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  $7 \times 7$  pixel image.

Because the integral over all pixels is done in frequency space, the maximum display size is determined by the smallest increment of spatial frequency in the grid of OTF's. Since the OTF varies over the field of view, ZEMAX computes all the OTF's, then scales and interpolates all of them to a common scale. The maximum display size can be increased by increasing the OTF sampling. The maximum display size is approximately given by  $\lambda F n$ , where  $\lambda$  is the wavelength,  $F$  is the smallest F number in the field of view, and  $n$  is the number of points in the OTF sampling. For any given optical system, the maximum display size increases only with OTF sampling.

### IMA/BIM File Viewer

#### *Purpose:*

This feature displays IMA/BIM files without processing. See "Geometric Image Analysis" on page 142 for a discussion of IMA/BIM files.

#### *Settings:*

Item	Description
File	The name of the .IMA or .BIM image file. This file must reside in the \ImaFiles directory. 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:*

See "Geometric Image Analysis" on page 142 for a discussion of IMA/BIM files.

## **Biocular Analysis**

### Field of View

#### *Purpose:*

Displays the field of view for up to four configurations. See the discussion for important assumptions before using this feature.

## Settings:

Item	Description
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 represent 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

### Purpose:

Displays the dipvergence and convergence for biocular analysis.

## Settings:

Item	Description
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" on page 153.

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" on page 153 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.

## **Miscellaneous**

### **Field Curvature/Distortion**

#### ***Purpose:***

Displays the field curvature and distortion plots.

#### ***Settings:***

Item	Description
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 in percent. 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.
Ignore Vignetting Factors	See the discussion section.
Distortion	Select Standard, F-Theta, or Calibrated. See the discussion section for details.
Scan Type	Choose +y, +x, -y, or -x field scan direction.

#### ***Discussion:***

The field curvature plot shows the distance from the currently defined focal (image) plane to the paraxial focal plane as a function of field coordinate. The tangential data are the distances measured along the Z-axis from the currently defined focal plane to the paraxial focal plane 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 radial field.

The field curvature for the tangential and sagittal rays is defined as the distance from the defined image plane to the paraxial focus for that ray. In non-rotational systems, the real ray and the chief ray may never intersect, and so the value presented is at the point of closest approach.

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.

Users often ask why the field curvature plot does not always start at zero for zero field. The reason is that the plot shows the distance from the currently defined image plane to the paraxial focal plane, and the currently defined image plane need not be coincident with the paraxial focal plane. If there is any defocus, then the two planes are offset, and so is the field curvature data.

"Standard" 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 plane). 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. 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  $\theta$  is the angle in object space. ZEMAX actually uses the generalization described above, rather than this equation, however the idea is the undistorted height goes as the tangent of the field angle.

"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  $\theta$  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 is similar to "F-Theta" distortion, except that the "best-fit" focal length is used rather than the system focal length. Calibrated distortion measures the deviation from linearity between the image height and the field angle, 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. The reference height for the undistorted ray is given by

$$y_{ref} = f'\theta,$$

where  $f'$  is the calibrated focal length and  $\theta$  is the angle in object space. One somewhat surprising 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  $y_{ref} = f'\theta$ , 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" on page 158.

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.

ZEMAX cannot compute the distortion 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 ray trace to find the exact field coordinate to hit the desired image coordinate; implicitly removing the distortion. As a workaround, ZEMAX automatically changes the field type from real to paraxial image height for the purposes of this computation, and this may affect the accuracy of the results for some systems. Generally the results will be accurate if the distortion is small.

## Grid Distortion

### *Purpose:*

Displays a grid of chief ray intercept points to indicate distortion.

### *Settings:*

Item	Description
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.

### *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 plane 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  $x_p$  and  $y_p$  are the predicted image coordinates relative to a reference image point and  $f_x$  and  $f_y$  are the linear coordinates on the object surface relative to a reference point. For optical systems using angles as a field definition,  $f_x$  and  $f_y$  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 the object height is linear with field angle and not angle directly, 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  $\theta_r$  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 plane 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

$$P = 100\% \frac{R_{\text{distorted}}}{R_{\text{predicted}}}, \text{ where}$$

$$R_{\text{real}} = \sqrt{(x_r)^2 + (y_r)^2},$$

$$R_{\text{predicted}} = \sqrt{(x_p)^2 + (y_p)^2}, \text{ and}$$

$$R_{\text{distorted}} = \sqrt{(x_p - x_r)^2 + (y_p - y_r)^2},$$

and the subscripts r and p refer to the real and predicted coordinates on the image plane relative to the reference field position image location, respectively. Because the values for  $R_{\text{real}}$  and  $R_{\text{predicted}}$  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  $R_{\text{real}}$  is less than  $R_{\text{predicted}}$ , then the sign of  $P$  is changed to be negative.



***This definition of generalized distortion may not be applicable in all cases, and the results should be used with caution.***

ZEMAX cannot compute the distortion 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 ray trace to find the exact field coordinate to hit the desired image coordinate; implicitly removing the distortion. As a workaround, ZEMAX automatically changes the field type from real to paraxial image height for the purposes of this computation, and this may affect the accuracy of the results for some systems. Generally the results will be accurate if the distortion is small.

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" on page 96). This may or may not be different than the orientation of a spot diagram or other plot plotted in image space units.

## Footprint Diagram

### *Purpose:*

Displays the footprint of the beam superimposed on any surface. Used for showing the effects of vignetting and for checking surface apertures.

### *Settings:*

Item	Description
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.
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 draw all configurations at once, or select the one configuration to draw, or select "Current" to show the active configuration.
Color Rays By	Select "Fields" to use color to distinguish between each field position, or "Waves" to distinguish between each wavelength, or "Config" to distinguish between configurations.

### *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.

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 shown as a percentage.

## Longitudinal Aberration

### *Purpose:*

Displays the longitudinal aberration as a function of pupil height at each wavelength.

### *Settings:*

Item	Description
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.

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**

#### *Purpose:*

Displays the lateral color as a function of field height.

#### *Settings:*

Item	Description
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.

### *Discussion:*

This feature computes the lateral color either of two ways:

If "All wavelengths" is checked: The data computed is the distance on the image surface from the intercept of the chief ray at each wavelength to the primary wavelength chief ray intercept.

If "All Wavelengths" is not checked: The data computed is the distance on the image surface from the shortest wavelength chief ray intercept to the longest wavelength chief ray intercept.

The base of the plot is on axis, and the top of the plot represents the maximum field radius. Only positive field angles or heights in the Y direction are used. The vertical scale is always normalized to the maximum field angle or height. The horizontal scale is in micrometers. Either real or paraxial rays may be used.

If "Show Airy Disk" is check on, then the Airy disk radius will be approximated as the square root of the sum of the squares of the X and Y direction Airy disk radii.

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.

### **Y-Ybar Diagram**

#### *Purpose:*

Y-Ybar diagram.

#### *Settings:*

Item	Description
First Surface	The first surface for which data will be plotted.
Last Surface	The last surface for which data will be plotted.

Item	Description
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.

**Discussion:**

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.

**Chromatic Focal Shift**

**Purpose:**

Chromatic focal shift plot.

**Settings:**

Item	Description
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 the image plane 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  $\lambda$  is the primary wavelength.

**Dispersion Diagram**

**Purpose:**

Plots index of refraction as a function of wavelength for any material in the glass catalog.

**Settings:**

Item	Description
Min Wavelength	Defines the left X- axis of the dispersion plot.
Max Wavelength	Defines the right X- axis of the dispersion plot.
Minimum Index	Defines the bottom Y- axis of the dispersion plot. Enter zero for automatic scaling.
Maximum Index	Defines the top Y- axis of the dispersion plot. Enter zero for automatic scaling.
Glass	The name of the material.
Use Temperature, Pressure	If checked, then the change in index due to temperature and pressure effects will be considered.

**Discussion:**

This is useful for checking if constants of dispersion or other formula data were entered correctly.

## Glass Map

### *Purpose:*

Plots names of glasses on the glass 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 for glasses within the boundary values specified in the following table.

### *Settings:*

Item	Description
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.

### *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.

## Internal Transmittance vs. Wavelength

### *Purpose:*

Plots the internal transmittance for any thickness as a function of wavelength for any material in the glass catalog.

### *Settings:*

Item	Description
Min Wavelength	Defines the left X- axis of the plot.
Max Wavelength	Defines the right X- axis of the plot.
Minimum Transmission	Defines the bottom Y- axis of the plot.
Maximum Transmission	Defines the top Y- axis of the plot. Enter zero for automatic scaling.
Glass	The name of the material.
Thickness	The thickness of the glass in millimeters.

### *Discussion:*

This is useful for checking the transmittance of a particular glass. See the Chapter "Polarization Analysis".

## System Summary Graphic

### *Purpose:*

Displays in a graphic window a summary of the system data, similar to the text based system data report.

### *Settings:*

None.

### *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 the Reports Menu chapter.

## **Aberration coefficients**

### **Seidel Coefficients**

#### *Purpose:*

Displays Seidel (unconverted, transverse, and longitudinal), and wavefront aberration coefficients.

#### *Settings:*

Item	Description
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 standard surfaces. Any systems which contain coordinate breaks, gratings, paraxial, or other non-standard surfaces are not adequately described by the paraxial rays which are used to compute the coefficients.

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 symbols  $n$  and  $u$  refer to the index of refraction and the paraxial marginal ray angle in the object space side of each surface. The primes above the  $n$  and  $u$  symbols indicate that these are the values on the image space side of the surface. 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" on page 31.

## INTERRELATIONSHIP OF ABERRATION COEFFICIENTS

Name	Seidel	Wave	Description	Transverse	Longitudinal
Spherical	$S_1$	$\frac{S_1}{8}$	Spherical	$-\frac{S_1}{2n'u'}$	$\frac{S_1}{2n'u'^2}$
Coma	$S_2$	$\frac{S_2}{2}$	Sagittal	$-\frac{S_2}{2n'u'}$	$\frac{S_2}{2n'u'^2}$
			Tangential	$-\frac{3S_2}{2n'u'}$	$\frac{3S_2}{2n'u'^2}$
Astigmatism	$S_3$	$\frac{S_3}{2}$	From tangential to sagittal foci	$-\frac{S_3}{n'u'}$	$\frac{S_3}{n'u'^2}$
Field Curvature	$S_4$	$\frac{S_4}{4}$	Gaussian to Petzval	$-\frac{S_4}{2n'u'}$	$\frac{S_4}{2n'u'^2}$
	$S_3 + S_4$	$\frac{S_3 + S_4}{4}$	Gaussian to sagittal	$-\frac{(S_3 + S_4)}{2n'u'}$	$\frac{S_3 + S_4}{2n'u'^2}$
	$2S_3 + S_4$	$\frac{2S_3 + S_4}{4}$	Gaussian to medial	$-\frac{(2S_3 + S_4)}{2n'u'}$	$\frac{2S_3 + S_4}{2n'u'^2}$
	$3S_3 + S_4$	$\frac{3S_3 + S_4}{4}$	Gaussian to tangential	$-\frac{(3S_3 + S_4)}{2n'u'}$	$\frac{3S_3 + S_4}{2n'u'^2}$
Distortion	$S_5$	$\frac{S_5}{2}$	Distortion	$-\frac{S_5}{2n'u'}$	$\frac{S_5}{2n'u'^2}$
Axial Color	$C_L$	$\frac{C_L}{2}$	Chromatic aberrations are measured between the extreme defined wavelengths, referenced to the selected wavelength.	$\frac{C_L}{n'u'}$	$\frac{C_L}{n'u'^2}$
Lateral Color	$C_T$	$C_T$		$\frac{C_T}{n'u'}$	$\frac{C_T}{n'u'^2}$

The Petzval radius of curvature shown on the Seidel screen is in system lens units, as is the Lagrange (or optical) invariant.

### Zernike Fringe Coefficients

*Purpose:*

Calculates the Zernike coefficients using the Fringe (also called the "University of Arizona") polynomials.

## Settings:

Item	Description
Sampling	Specify the density in the pupil to use for coefficient fitting. Larger grid sizes are more accurate, although the computation time increases.
Max 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" on page 109.

## Discussion:

The Zernike screen displays the individual 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,  $\sigma$ , 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 affect 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" on page 47). 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" on page 133 for more information.

### 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  $\phi$  is measured counter clockwise from the local +x axis. The radial coordinate is the normalized dimensionless parameter  $\rho$ .

#### ZERNIKE FRINGE POLYNOMIALS

Term	$Z(\rho, \phi)$
1	1
2	$\rho \cos \phi$
3	$\rho \sin \phi$
4	$2\rho^2 - 1$
5	$\rho^2 \cos 2\phi$
6	$\rho^2 \sin 2\phi$
7	$(3\rho^2 - 2)\rho \cos \phi$
8	$(3\rho^2 - 2)\rho \sin \phi$
9	$6\rho^4 - 6\rho^2 + 1$
10	$\rho^3 \cos 3\phi$
11	$\rho^3 \sin 3\phi$
12	$(4\rho^2 - 3)\rho^2 \cos 2\phi$
13	$(4\rho^2 - 3)\rho^2 \sin 2\phi$
14	$(10\rho^4 - 12\rho^2 + 3)\rho \cos \phi$
15	$(10\rho^4 - 12\rho^2 + 3)\rho \sin \phi$

Term	$Z(\rho, \varphi)$
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	$(15\rho^4 - 20\rho^2 + 6)\rho^2 \cos 2\varphi$
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

*Purpose:*



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.

*Settings:*

Item	Description
Sampling	Specify the density in the pupil to use for coefficient fitting. Larger grid sizes are more accurate, although the computation time increases.
Max 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" on page 109.

*Discussion:*

This feature is almost identical to the "Zernike Fringe Coefficients" feature, except 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" on page 165 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  $\rho^5$ . Only certain values for m are allowed once n is chosen; n + m must be even, and  $0 \leq m \leq n$ . 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  $\phi$  is measured counter clockwise from the local +x axis. The radial coordinate is the normalized dimensionless parameter  $\rho$ .

# 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	$\sqrt{7}(20\rho^6 - 30\rho^4 + 12\rho^2 - 1)$

Term	$Z(\rho, \phi)$
23	$\sqrt{14}(15\rho^6 - 20\rho^4 + 6\rho^2)\sin 2\phi$
24	$\sqrt{14}(15\rho^6 - 20\rho^4 + 6\rho^2)\cos 2\phi$
25	$\sqrt{14}(6\rho^6 - 5\rho^4)\sin 4\phi$
26	$\sqrt{14}(6\rho^6 - 5\rho^4)\cos 4\phi$
27	$\sqrt{14}\rho^6\sin 6\phi$
28	$\sqrt{14}\rho^6\cos 6\phi$

### Zernike Annular Coefficients

#### *Purpose:*

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.

#### *Settings:*

Item	Description
Sampling	Specify the density in the pupil to use for coefficient fitting. Larger grid sizes are more accurate, although the computation time increases.
Max 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, $\varepsilon$ 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" on page 109.

#### *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  $\varepsilon$ , defined as

$$\varepsilon = \frac{R_{inner}}{R_{outer}},$$

where Rinner and Router are the inner and outer radii of the annular pupil. When  $\epsilon$  is zero, the annulus is a circle, and the Annular and Standard Zernike polynomials are identical. When  $\epsilon$  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  $\phi$  is measured counter clockwise from the local +x axis. The radial coordinate is the normalized dimensionless parameter  $\rho$ .

### ZERNIKE ANNULAR POLYNOMIALS

Term	$Z(\rho, \phi, \epsilon)$
1	1
2	$\frac{2\rho \cos \phi}{\sqrt{1 + \epsilon^2}}$
3	$\frac{2\rho \sin \phi}{\sqrt{1 + \epsilon^2}}$
4	$\frac{\sqrt{3}(1 + \epsilon^2 - 2\rho^2)}{\epsilon^2 - 1}$
5	$\frac{\sqrt{6}\rho^2 \sin 2\phi}{\sqrt{1 + \epsilon^2 + \epsilon^4}}$
6	$\frac{\sqrt{6}\rho^2 \cos 2\phi}{\sqrt{1 + \epsilon^2 + \epsilon^4}}$
7	$\frac{2\sqrt{2}\rho(-2 - 2\epsilon^4 + 3\rho^2 + \epsilon^2(-2 + 3\rho^2)) \sin \phi}{\sqrt{(\epsilon^2 - 1)^2(1 + \epsilon^2)(1 + 4\epsilon^2 + \epsilon^4)}}$
8	$\frac{2\sqrt{2}\rho(-2 - 2\epsilon^4 + 3\rho^2 + \epsilon^2(-2 + 3\rho^2)) \cos \phi}{\sqrt{(\epsilon^2 - 1)^2(1 + \epsilon^2)(1 + 4\epsilon^2 + \epsilon^4)}}$
9	$\frac{2\sqrt{2}\rho^3 \sin 3\phi}{\sqrt{1 + \epsilon^2 + \epsilon^4 + \epsilon^6}}$
10	$\frac{2\sqrt{2}\rho^3 \cos 3\phi}{\sqrt{1 + \epsilon^2 + \epsilon^4 + \epsilon^6}}$
11	$\frac{\sqrt{5}(1 + \epsilon^4 - 6\rho^2 + 6\rho^4 + \epsilon^2(4 - 6\rho^2))}{(\epsilon^2 - 1)^2}$

Term	$Z(\rho, \varphi, \varepsilon)$
12	$\frac{\sqrt{10} \sqrt{\frac{(\varepsilon^2 - 1)^4 (1 + \varepsilon^2 + \varepsilon^4)}{1 + 4\varepsilon^2 + 10\varepsilon^4 + 4\varepsilon^6 + \varepsilon^8}} \rho^2 (3 + 3\varepsilon^6 - 4\rho^2 + \varepsilon^2(3 - 4\rho^2) + \varepsilon^4(3 - 4\rho^2)) \cos 2\varphi}{(\varepsilon^2 - 1)^3 (1 + \varepsilon^2 + \varepsilon^4)}$
13	$\frac{\sqrt{10} \sqrt{\frac{(\varepsilon^2 - 1)^4 (1 + \varepsilon^2 + \varepsilon^4)}{1 + 4\varepsilon^2 + 10\varepsilon^4 + 4\varepsilon^6 + \varepsilon^8}} \rho^2 (3 + 3\varepsilon^6 - 4\rho^2 + \varepsilon^2(3 - 4\rho^2) + \varepsilon^4(3 - 4\rho^2)) \sin 2\varphi}{(\varepsilon^2 - 1)^3 (1 + \varepsilon^2 + \varepsilon^4)}$
14	$\frac{\sqrt{10} \rho^4 \cos 4\varphi}{\sqrt{1 + \varepsilon^2 + \varepsilon^4 + \varepsilon^6 + \varepsilon^8}}$
15	$\frac{\sqrt{10} \rho^4 \sin 4\varphi}{\sqrt{1 + \varepsilon^2 + \varepsilon^4 + \varepsilon^6 + \varepsilon^8}}$
16	$\frac{(-2\sqrt{3})EQ\rho \cos \varphi}{(\varepsilon^2 - 1)^3 (1 + 4\varepsilon^2 + \varepsilon^4)}, \text{ where}$ $E = \sqrt{\frac{(\varepsilon^2 - 1)^2 (1 + 4\varepsilon^2 + \varepsilon^4)}{1 + 9\varepsilon^2 + 9\varepsilon^4 + \varepsilon^6}}, \text{ and}$ $Q = (3 + 3\varepsilon^8 - 12\rho^2 + 10\rho^4 - 12\varepsilon^6(\rho^2 - 1) + 2\varepsilon^4(15 - 24\rho^2 + 5\rho^4) + 4\varepsilon^2(3 - 12\rho^2 + 10\rho^4)).$
17	$\frac{(-2\sqrt{3})EQ\rho \sin \varphi}{(\varepsilon^2 - 1)^3 (1 + 4\varepsilon^2 + \varepsilon^4)}, \text{ where E and Q are defined as in term 16 above.}$
18	$\frac{(-2\sqrt{3})EQ\rho^3 \cos 3\varphi}{(\varepsilon^2 - 1)^4 (1 + \varepsilon^2)(1 + \varepsilon^4)}, \text{ where}$ $E = \sqrt{\frac{(\varepsilon^2 - 1)^6 (1 + \varepsilon^2)(1 + \varepsilon^4)}{1 + 4\varepsilon^2 + 10\varepsilon^4 + 20\varepsilon^6 + 10\varepsilon^8 + 4\varepsilon^{10} + \varepsilon^{12}}}, \text{ and}$ $Q = 4 + 4\varepsilon^8 - 5\rho^2 + \varepsilon^2(4 - 5\rho^2) + \varepsilon^4(4 - 5\rho^2) + \varepsilon^6(4 - 5\rho^2).$
19	$\frac{(-2\sqrt{3})EQ\rho^3 \sin 3\varphi}{(\varepsilon^2 - 1)^4 (1 + \varepsilon^2)(1 + \varepsilon^4)}, \text{ where E and Q are defined as in term 18 above.}$
20	$\frac{2\sqrt{3} \rho^5 \cos 5\varphi}{\sqrt{1 + \varepsilon^2 + \varepsilon^4 + \varepsilon^6 + \varepsilon^8 + \varepsilon^{10}}}$

Term	$Z(\rho, \varphi, \epsilon)$
21	$\frac{2\sqrt{3}\rho^5 \sin 5\varphi}{\sqrt{1 + \epsilon^2 + \epsilon^4 + \epsilon^6 + \epsilon^8 + \epsilon^{10}}}$
22	$\frac{\sqrt{7}(1 + \epsilon^6 - 12\rho^2 + 30\rho^4 - 20\rho^6 - 3\epsilon^4(-3 + 4\rho^2) + 3\epsilon^2(3 - 12\rho^2 + 10\rho^4))}{(\epsilon^2 - 1)^3}$

## **Calculations**

### **Ray Trace**

#### *Purpose:*

Paraxial and real trace of a single ray.

#### *Settings:*

Item	Description
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.

### **Fiber Coupling Efficiency**

#### *Purpose:*

This feature computes the coupling efficiency for single mode fiber coupling systems. For multi-mode fiber coupling, see “Calculating efficiency of multi-mode fibers” on page 145. Also see “Computing Fiber Coupling” on page 530.

**Settings:**

Item	Description
Source Fiber	
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 illumination of the pupil as specified by the System, General, Apodization setting will be used. All computations are then referenced to the total energy incident upon the entrance pupil.
Receiving Fiber	
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 <u>after</u> 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 <u>before</u> 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, polarized rays will be traced for each ray required, and the resulting transmitted intensity through the system will be accounted for. See the “System Menu” chapter under “Polarization” for information on defining the polarization state and other details. Only ZEMAX-EE supports this capability.

**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” on page 145. For a more detailed and flexible physical optics treatment of fiber coupling see “Computing Fiber Coupling” on page 530.

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 partially 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" on page 86).

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:

$$S = \left[ \frac{\iint t(x, y) F_s(x, y) dx dy}{\iint F_s(x, y) dx dy} \right]^2,$$

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:

$$T = \frac{\left| \iint F_r(x, y) W'(x, y) dx dy \right|^2}{\iint F_r(x, y) F_r'(x, y) dx dy \iint W(x, y) W'(x, y) dx dy},$$

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 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.

Large phase errors in the wavefront function are difficult to numerically integrate accurately. Fortunately, rays with huge phase aberrations are not going to contribute to the fiber coupling anyway, as these rays move through  $2\pi$  phase cycles so quickly that they tend to all average to zero. To improve the accuracy of the fiber coupling computation for highly aberrated systems, the integration algorithm will ignore rays with more than  $n/8$  waves of OPD error, where  $n$  is the number of rays across the pupil, which is determined by the sampling setting.

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.

## YNI Contributions

### *Purpose:*

This feature lists for each surface the paraxial YNI value, which is proportional to the Narcissus contribution of that surface.



### Settings:

Item	Description
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.

## Sag Table

### Purpose:

This feature lists for the selected surface the sag (z-coordinate) 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. Only the Y-coordinate of the surface is considered, therefore this feature is not useful, and should not be used, if the surface is non-rotationally symmetric.

### Settings:

Item	Description
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".

### 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 five columns:

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.

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.

It is always advisable to check the data and sign convention carefully before any fabrication decisions are made.

## **Cardinal Points**

### ***Purpose:***

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.

### ***Settings:***

Item	Description
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.

## **Glass and Gradient Index**

### **Dispersion Diagram**

### ***Purpose:***

Plots index of refraction as a function of wavelength for any material in the glass catalog.

### ***Settings:***

Item	Description
Min Wavelength	Defines the left X- axis of the dispersion plot.
Max Wavelength	Defines the right X- axis of the dispersion plot.
Minimum Index	Defines the bottom Y- axis of the dispersion plot. Enter zero for automatic scaling.
Maximum Index	Defines the top Y- axis of the dispersion plot. Enter zero for automatic scaling.
Glass	The name of the material.
Use Temperature, Pressure	If checked, then the change in index due to temperature and pressure effects will be considered.

### ***Discussion:***

This is useful for checking if constants of dispersion or other formula data were entered correctly.

## **Glass Map**

### ***Purpose:***

Plots names of glasses on the glass map according to the index of refraction and Abbe V-number. The index and Abbe number are taken directly from the entries in the glass catalog, and are not computed based upon the wavelength data or dispersion coefficients. All currently loaded glass catalogs are searched for glasses within the boundary values specified in the following table.

### *Settings:*

Item	Description
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.

### *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.

## Internal Transmittance vs. Wavelength

### *Purpose:*

Plots the internal transmittance for any thickness as a function of wavelength for any material in the glass catalog.

### *Settings:*

Item	Description
Min Wavelength	Defines the left X- axis of the plot.
Max Wavelength	Defines the right X- axis of the plot.
Minimum Transmission	Defines the bottom Y- axis of the plot.
Maximum Transmission	Defines the top Y- axis of the plot. Enter zero for automatic scaling.
Glass	The name of the material.
Thickness	The thickness of the glass in millimeters.

### *Discussion:*

This is useful for checking the transmittance of a particular glass. See the Chapter “Polarization Analysis”.

## Grin Profile

### *Purpose:*

Plots the index of refraction along any axis for gradient index surfaces.

### *Settings:*

Item	Description
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.

Item	Description
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**

**Purpose:**

Plots the axial index profile of a Gradium surface.

**Settings:**

Item	Description
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.

### **Universal Plot**



***This feature is only available in the EE edition of ZEMAX.***

### **New Universal Plot...**

**Purpose:**

Displays as either a plot or as a text listing the value of any optimization operand as a function of some other parameter.

**Settings:**

Item	Description
Independent variable 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.

Item	Description
	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 settings	
Operand	The optimization operand to use as the dependent variable function. The whole system merit function is also included at the top of the list. 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. The whole system merit function is also included at the top of the list.
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.
Hx/Hy/Px/Py	The Hx, Hy, Px, Py values for the selected operand. The dialog box will display the short 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 (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 directory, ZEMAX will only recognize the file if it resides in the main ZEMAX directory. 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 (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 any number of system, surface, or multi-configuration data.

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 the Chapter "Optimization" 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".

The escape key will terminate the analysis if the computation is taking too long.

## **Polarization**



***These features are only available in the EE editions of ZEMAX.***

---

For more information on polarization calculations, see "POLARIZATION ANALYSIS" on page 497.

### **Polarization Ray Trace**

#### ***Purpose:***

The polarization ray trace feature generates a text window which displays all the polarization data for a single ray.

#### ***Settings:***

Item	Description
Jx, Jy	Jones electric field. see "Defining the incident polarization" on page 510.
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.

#### ***Discussion:***

This feature tabulates all of the data computed by ZEMAX to perform polarization ray tracing. See "POLARIZATION ANALYSIS" on page 497.

### **Polarization Pupil Map**

#### ***Purpose:***

Generates a graph of the polarization ellipse as a function of pupil position. This aids in visualizing the change in polarization over the pupil.

#### ***Settings:***

Item	Description
Jx, Jy	Jones electric field. see "Defining the incident polarization" on page 510.
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.

Item	Description
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.

*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” on page 497.

**Transmission**

*Purpose:*

Computes the integrated and surface by surface transmission through the optical system considering polarization effects.

*Settings:*

Item	Description
Jx, Jy	Jones electric field. see “Defining the incident polarization” on page 510.
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” on page 139.

**Phase Aberration**

*Purpose:*

Computes the polarization induced phase aberration of an optical system.

*Settings:*

Item	Description
Jx, Jy	Jones electric field. see “Defining the incident polarization” on page 510.
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:*

This feature computes for the specified field position and wavelength the phase aberration in the image space X and Y orientations. Polarization phase aberrations are induced by the effects of refraction through dielectric media, and by reflection from metallic or dielectric mirrors.

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

### *Purpose:*

Generates a graph of the transmitted intensity for each field and wavelength as a function of either tangential or sagittal pupil fans.

### *Settings:*

Item	Description
Jx, Jy	Jones electric field. see "Defining the incident polarization" on page 510.
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" on page 497.

## Coatings



***These features are only available in the EE editions of ZEMAX.***

For more information on coating calculations, see "POLARIZATION ANALYSIS" on page 497.

## Reflection vs. Angle

### *Purpose:*

Computes the S, P, and average polarization intensity coefficients for reflection for the specified surface as a function of incident angle.

### *Settings:*

Item	Description
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.



Item	Description
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.
Coat Group	The coat/scatter group (CSG) 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**

*Purpose:*

Computes the S, P and average polarization intensity coefficients for transmission for the specified surface as a function of incident angle.

*Settings:*

See "Reflectivity vs. Angle" above.

*Discussion:*

The incident angle is measured in the medium prior to the specified surface.

**Absorption vs. Angle**

*Purpose:*

Computes the S, P, and average polarization intensity coefficients for absorption for the specified surface as a function of incident angle.

*Settings:*

See "Reflectivity vs. Angle" above.

*Discussion:*

The incident angle is measured in the medium prior to the specified surface.

**Diattenuation vs. Angle**

*Purpose:*

Computes the R (reflected) and T (transmitted) diattenuation for the specified surface as a function of incident angle.

*Settings:*

See "Reflectivity vs. Angle" above.

*Discussion:*

The incident angle is measured in the medium prior to the specified surface.

**Phase vs. Angle**

*Purpose:*

Computes the S and P polarization phase for reflection (if the surface is a mirror) or for transmission (if the surface is not a mirror) for the specified surface as a function of incident angle.

*Settings:*

See "Reflectivity vs. Angle" above.

*Discussion:*

The incident angle is measured in the medium prior to the specified surface.

**Retardance vs. Angle**

*Purpose:*

Computes the retardance for the specified surface as a function of incident angle.

*Settings:*

See "Reflectivity vs. Angle" above.

*Discussion:*

The incident angle is measured in the medium prior to the specified surface.

**Reflection vs. Wavelength**

*Purpose:*

Computes the S, P, and average polarization intensity coefficients for reflection for the specified surface as a function of incident wavelength.

*Settings:*

Item	Description
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.
Coat Group	The coat/scatter group (CSG) 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" on page 497.

**Transmission vs. Wavelength**

*Purpose:*

Computes the S, P, and average polarization intensity coefficients for transmission for the specified surface as a function of incident wavelength.

*Settings:*

See "Reflectivity vs. Wavelength" above.

*Discussion:*

See "POLARIZATION ANALYSIS" on page 497.

**Absorption vs. Wavelength**

*Purpose:*

Computes the S, P, and average polarization intensity coefficients for absorption for the specified surface as a function of incident wavelength.

### *Settings:*

See "Reflectivity vs. Wavelength" above.

### **Diattenuation vs. Wavelength**

#### *Purpose:*

Computes the R (reflected) and T (transmitted) diattenuation for the specified surface as a function of incident wavelength.

#### *Settings:*

See "Reflectivity vs. Wavelength" above.

### **Phase vs. Wavelength**

#### *Purpose:*

Computes the S and P polarization phase for reflection (if the surface is a mirror) or for transmission (if the surface is not a mirror) for the specified surface as a function of incident wavelength.

#### *Settings:*

See "Reflectivity vs. Wavelength" above.

### **Retardance vs. Wavelength**

#### *Purpose:*

Computes the retardance for the specified surface as a function of incident wavelength.

#### *Settings:*

See "Reflectivity vs. Wavelength" above.

## **Physical Optics**

### **Paraxial Gaussian Beam**

#### *Purpose:*

Computes Paraxial Gaussian Beam parameters. Paraxial Gaussian Beams are limited to axial analysis of rotationally symmetric systems. See also "Skew Gaussian Beam" on page 189 and "Physical Optics Propagation" on page 190.

#### *Settings:*

Item	Description
Wavelength	The wavelength number to use for the calculation.
M <sup>2</sup> Factor	The M <sup>2</sup> 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	
Surface	

#### *Discussion:*

This feature computes ideal and mixed mode M<sup>2</sup> 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" on page 190.

### 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  $M^2$ , which defines the relative beam size and divergence with respect to the embedded Gaussian mode. The ideal  $M^2$  value is unity, but real lasers will always have an  $M^2$  value greater than unity.

This feature requires the definition of the initial input embedded beam properties, and the  $M^2$  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  $\omega_0$  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:

$$\phi = \tan^{-1} \frac{z}{z_r}$$

### 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  $M^2$ . The factor  $M^2$  can be thought of as "times diffraction limited" number, and is always greater than unity. The  $M^2$  factor determines the size of the real, aberrated Gaussian beam by scaling the size and divergence of the embedded Gaussian mode by  $M$ . It is common practice to specify  $M^2$  for a laser beam, rather than  $M$ , although the factor  $M$  is used to scale the beam size. The  $M^2$  factor must be measured to be determined correctly. If the  $M^2$  factor is set to unity, the default value, ZEMAX computes the TEM00 data described above. If  $M^2$  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

#### *Purpose:*

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" on page 187 and "Physical Optics Propagation" on page 190.

#### *Settings:*

Item	Description
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.

Item	Description
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" on page 187.

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  $p_x = 0$ ,  $p_y = 1$  marginal ray coordinate at the entrance pupil. The  $p_x = 1$ ,  $p_y = 0$  marginal ray defines the +x direction. For a discussion of the normalized pupil coordinates, see "Normalized field and pupil coordinates" on page 49.

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.

### Physical Optics Propagation



***This feature is only available in the EE editions of ZEMAX.***

#### *Purpose:*

This feature propagates an arbitrary coherent optical beam through the optical system. For a complete description of this feature see "PHYSICAL OPTICS PROPAGATION" on page 515. The Physical Optics Propagation feature results are generally not accurate for systems including non-sequential component groups; see "Propagating through non-sequential surfaces" on page 525 for more information. For the most accurate results, replace non-sequential component groups with sequential equivalents.

#### *Settings:*

##### *Beam Definition Tab:*

Item	Description
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" on page 524.

Item	Description
Wavelength	The wavelength number to use for the beam.
Field	The field number of the chief ray used to align the initial beam.
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.
Type	See the discussion and "Defining the initial beam" on page 526.
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.
File	See the discussion and "Defining the initial beam" on page 526.
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.
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.
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" on page 526.

*Display Tab:*

Item	Description
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, and contains warnings that indicate a possible inaccuracy 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.

Item	Description
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 directory \POP\BeamFiles.
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 directory \POP\BeamFiles.
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.
Normalize Cross Sections To	If zero, the vertical scale on cross section plots will be set by the maximum data value. Otherwise, the maximum vertical scale is set by the normalization value or the maximum data value, whichever is larger. 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" on page 131. 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:*

Item	Description
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" on page 530.
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" on page 530.
File	Selects the name of the DLL or data file describing the fiber mode. See "Computing Fiber Coupling" on page 530.
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" on page 515.

#### *Comment 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 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" on page 193.

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" on page 87.

Total Power: The total power in the beam. To set the power units see "Units" on page 87.

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" on page 520.

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" on page 530.

## **Beam File Viewer**



***This feature is only available in the EE editions of ZEMAX.***

---

### ***Purpose:***

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" on page 190. See also the Chapter "PHYSICAL OPTICS PROPAGATION" on page 515. For information on the ZBF file format, see "ZEMAX Beam File (ZBF) binary format" on page 528.

### ***Settings:***

The available settings are a subset of what is available for the Physical Optics Propagation feature described above.

### ***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.



***For rapid viewing through a range of files that differ only by the surface number, use the left and right cursor keys.***

---

See the "Comment about beam projection" on page 192.

ZBF files are stored in the directory \POP\BeamFiles.



## **Optimization**

### **Optimization**

#### *Purpose:*

Optimization is used to improve or modify a design to meet specific conditions.

#### *Discussion:*

See "OPTIMIZATION" on page 385 for a full discussion of building merit functions, setting variables, and using the optimization feature.

### **Global Search**

#### *Purpose:*

This feature initiates a search for the global optimum, which is the best possible design for a given merit function and set of variables.

#### *Discussion:*

See "GLOBAL OPTIMIZATION" on page 435.

### **Hammer Optimization**

#### *Purpose:*

This feature automates the repetitive optimization of a design to escape local minima in the merit function.

#### *Discussion:*

See "GLOBAL OPTIMIZATION" on page 435.

### **Merit Function Listing**

#### *Purpose:*

This function generates a text listing of the merit function which can be saved or printed. The overall merit function numerical value is listed, as well as the numerical merit function of all user added operands (defined as those operands placed before the DMFS operand) and the default merit function (all operands after the DMFS operand). See also "Merit function definition" on page 391.

### **Remove All Variables**

#### *Purpose:*

This tool 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.

### **Glass Substitution Template**

#### *Purpose:*

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" on page 485), and when computing the value of the RGLA operand ("RGLA" on page 415).

## Settings:

Item	Description
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 No 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" on page 478.
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.
Maximum Climatic Resistance (CR)	The maximum allowed value of the CR code. Use zero to ignore CR.
Maximum Stain Resistance (FR)	The maximum allowed value of the FR code. Use zero to ignore FR.
Maximum Acid Resistance (SR)	The maximum allowed value of the SR code. Use zero to ignore SR.
Maximum Alkali Resistance (AR)	The maximum allowed value of the AR code. Use zero to ignore AR.
Maximum Phosphate Resistance (PR)	The maximum allowed value of the PR code. Use zero to ignore PR.
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.

## **Tolerancing**

### Tolerancing

#### *Purpose:*

Tolerancing.

#### *Discussion:*

See the chapter "Tolerancing" for a complete discussion.

### Tolerance Listing

#### *Purpose:*

This function generates a text listing of the tolerances which can be saved or printed.

### Tolerance Summary

#### *Purpose:*

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.

## **Test Plates**

### **Test Plate Fitting**

#### *Purpose:*

Provides automatic fitting of radii to a vendor test plate list.

#### *Settings:*

Item	Description
File Name	Selects different test plate lists
Method of Fit	Chooses the fitting order, which may affect the quality of the fit.
# Opt Cycles	Chooses the number of optimization cycles to run, or Automatic.

#### *Discussion:*

This feature automates the task of matching radii of lens elements to the existing tooling of a particular vendor. The current merit function is used as a figure of merit during the fitting process.

To match a particular radius, make it variable on the lens data editor. As many radii as desired can be matched at one time. ZEMAX will attempt to fit all radii with variable status in the Lens Data Editor, 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.

Now press OK and the test plate fitting will begin. ZEMAX starts by searching 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. 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 editor.

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 on the screen. 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 \TESTPLAT directory. The files are in ASCII with 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

### *Purpose:*

Displays in a text window a list of test plates from a specific vendor.

### *Settings:*

Item	Description
File Name	The name of the test plate file. The file must be in the \TESTPLAT directory.

### *Discussion:*

All units on the report are in mm. The CC and CX columns indicate the availability of concave and convex test plates, respectively.

## Catalogs

### Glass Catalogs

### *Purpose:*

Provides access to the glass catalogs.

### *Discussion:*

See the chapter "Using Glass Catalogs" for an extended discussion of this feature.

## Glass Compare

### *Purpose:*

Compares 2 or 3 glass catalogs to assist in finding glasses used in visible light with similar index and dispersion characteristics.

### *Settings:*

Item	Description
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.

## Lens Catalogs

### *Purpose:*

Used to search for a particular lens from or browse through the stock lens catalogs.

### *Settings:*

Item	Description
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 directory for stock lenses, which is set on the Directories tab of the File, Preferences dialog box.
Use Focal Length	If checked, then the specified 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 focal lengths in millimeters.
Use 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.
Max elements	Selects the maximum 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 directory 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 the Lens Data Editor. A prompt will be issued for the insert file options, such as at which surface the lens should be inserted.
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.
Exit	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 subdirectory within the ZEMAX directory called STOCKCAT. Within STOCKCAT, 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	Contact Data
ARCHEROPTX.ZMF	Archer OpTx 3412 Enterprise Drive, Rowlett, TX 75088 972-463-8001 www.archeroptx.com
COHERENT.ZMF	Coherent 3845 Atherton Road Suite #1, Rocklin, CA 95765. Tel: 916-435-4230 Fax: 916-435-4234 Web: www.ealingcatalog.com
CORNING.ZMF	Corning 3997 McMann Road, Cincinnati, OH 45245 Tel: 513-752-7000 Web: www.corningprecisionlens.com



File Name	Contact Data
CVI.ZMF	CVI 200 Dorado Place SE, Albuquerque, New Mexico 87123 Tel: 800-296-9541 Fax: 505-298-9908 Web: www.cvilaser.com
EDMUND OPTICS.ZMF	Edmund Industrial Optics 101 East Gloucester Pike, Barrington, NJ 08007 Tel: 856-573-6250 Fax: 856-573-6295 Web: www.edmundoptics.com
ESCO.ZMF	Esco Products 171 Oak Ridge Road, Oak Ridge, New Jersey 07438 Tel: 800-922-3726 Fax: 973-697-3011 Web: www.escoproducts.com
GELTECH.ZMF	LightPath Technologies, Inc. Polycarbonate (Geltech) lenses from LightPath Technologies (see below for contact data).
ISP OPTICS.ZMF	ISP Optics 1 Bridge Street, Irvington, NY 10533 Tel: 914-591-3070 Fax: 914-591-3715 Web: www.ispoptics.com
JML.ZMF	JML Optical Industries, Inc. 76 Fernwood Ave., Rochester, NY 14621 Tel: 585-342-8900 Fax: 585-342-6125 Web: www.jmloptical.com
LIGHTPATH.ZMF	LightPath Technologies, Inc. 2603 Challenger Tech Ct, Suite 100, Orlando, FL 32826 Tel: 407-382-4003 Fax: 407-382-4007 Web: www.lightpath.com
LINOS PHOTONICS.ZMF	LINOS Photonics, Inc. 459 Fortune Boulevard, Milford, MA 01757-1723 Tel: 508-478-6200 Fax: 508-478-5980 Web: www.linos-photonics.com
MELLES GRIOT.ZMF	Melles Griot 2051 Palomar Airport Rd., 200, Carlsbad, California 92009 Tel: 800-835-2626 Fax: 760-804-0049 Web: www.mellesgriot.com
NEWPORT CORP.ZMF	Newport Corporation 1791 Deere Avenue, Irvine, CA 92606 Tel: 800-222-6440 Fax: 949-253-1680 Web: www.newport.com
NSG.ZMF	NSG America, Inc. 28 World's Fair Drive, Somerset, NJ 08873 Tel: 732-469-9650 Fax: 732-469-9654 Web: www.nsgamerica.com

File Name	Contact Data
OPTICS FOR RESEARCH.ZMF	Optics For Research P.O. Box 82 Caldwell, NJ 07006 Tel: 973-228-4480 Fax: 973-228-0915 Web: www.ofr.com
OPTOSIGMA.ZMF	OptoSigma Corporation 2001 Deere Avenue, Santa Ana, California 92705 Tel: 949-851-5881 Fax: 949-851-5058 Web: www.optosigma.com
PHILIPS.ZMF	Philips Competence Centre Plastic B.V. PO Box 218-5600 MD, Eindhoven, The Netherlands Tel: 31-40-273-2394
ROLYN OPTICS.ZMF	Rolyn Optics Company 706 Arrowgrand Circle, Covina, California 91722 Tel: 626-915-5707 Fax: 626-915-1379 Web: www.rolyn.com
ROSS OPTICAL.ZMF	Ross Optical Industries 1410 Gail Borden, El Paso, TX 79935 Tel: 800-880-5417 Fax: 915-595-5466 Web: www.rossoptical.com
SPECIAL OPTICS.ZMF	Special Optics 315 Richard Mine Road, Wharton, NJ 07885 Tel: 973-366-7289 Fax: 973-366-7407 Web: www.specialoptics.com
THORLABS.ZMF	Thorlabs Inc. 435 Route 206 North, Newton, NJ 07860 Tel: 973-300-3007 Fax: 973-300-3607 Web: 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.

## **Coatings**

### **Edit Coating File**



***This feature is only available in the EE edition of ZEMAX.***

#### ***Purpose:***

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". 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



***This feature is only available in the EE edition of ZEMAX.***

---

### *Purpose:*

Reloads the coating file for the current lens. This is required if the file has been edited since the lens was loaded.

## Add Coatings to All Surfaces



***This feature is only available in the EE edition of ZEMAX.***

---

### *Purpose:*

Adds any specified coating to all surfaces with air-glass boundaries.

### *Discussion:*

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.

## Coating Listing



***This feature is only available in the EE edition of ZEMAX.***

---

### *Purpose:*

This function generates a text listing of the definitions of materials and coatings contained in the coating file for the current lens.

## Scattering

### ABg Scatter Data Catalogs

### *Purpose:*

Provides access to the ABg scatter data catalogs.

### *Settings:*

Item	Description
File	The current ABg scatter data file name. This file must be placed in the \ABG_DATA directory; the file name may be selected on the System, General dialog box on the Files tab, as described on page 91.
Name	The name of the ABg data being edited.
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. 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.

Item	Description
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, as described on page 91. 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” on page 354.

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:

$$A' = A \left( \frac{\lambda}{\lambda_{ref}} \right)^{g-4}, \text{ and}$$

$$B' = B \left( \frac{\lambda}{\lambda_{ref}} \right)^g,$$

where  $\lambda$  is the wavelength being traced and the unsubscripted A, B, and g parameters are measured at the reference wavelength,  $\lambda_{ref}$ .

### Scatter Function Viewer

#### *Purpose:*

Displays BSDF as a function of the magnitude of the scatter vector x.

## Settings:

Item	Description
Lambertian	Selects Lambertian scattering.
Gaussian	Selects Gaussian scattering.
ABg	Selects ABg scattering.
User Defined	Selects User Defined scattering.
Wavelength	The wavelength number to use. ABg scattering uses the specified wavelength, along with the reference wavelength defined in the ABg catalog to scale the ABg coefficients.
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 scatter functions.
Sigma	The sigma value to use for Gaussian scattering.
ABg Name	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” on page 203.
DLL Name	The name of the user defined scattering DLL to use. See “User defined scattering” on page 356.
(parameter names)	These parameters are for user defined scattering DLL.
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 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.
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” on page 353.

For User Defined and ABg scattering, the Theta X and Theta Y values are used to define the specular ray direction cosines. The defining equations which convert angles to cosines are as follows:

$$n = 1 / (\sqrt{1 + (\tan \alpha_x)^2 + (\tan \alpha_y)^2}),$$

$$l = n \tan(\alpha_x), \text{ and}$$

$$m = n \tan(\alpha_y).$$

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 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 total integrated scatter is always 1.0.

ABg: The ABg values are defined in the ABg Scatter Data Catalogs; see "ABg Scatter Data Catalogs" on page 203 for more information. The total integrated scatter is not generally 1.0; the actual value is listed in the output plot or listing. 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 uses the catalog data with the angle of incidence closest to that of the defined specular ray, however, no scaling or interpolation with angle is done.

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.

## **Apertures**

### **Convert Semi-Diameters to Circular Apertures**

#### *Purpose:*

Converts all surfaces without surface apertures to fixed semi-diameters with circular apertures set to the semi-diameter value.

#### *Discussion:*

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 the chapter "System Menu" for a description of 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**

#### *Purpose:*

Converts all surfaces without surface apertures to floating apertures which vignette at the semi-diameter value.

#### *Discussion:*

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.

### **Remove All Apertures**

#### *Purpose:*

Turns off all surface apertures.

#### *Discussion:*

All surface aperture types are set to "none" by this tool.

### **Replace Vignetting With Apertures**

#### *Purpose:*

Turns off vignetting factors and places surface apertures.

#### *Discussion:*

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.

## **Fold Mirrors**

### **Add Fold Mirror**

#### *Purpose:*

Inserts a fold mirror, including the required coordinate breaks, to bend a beam.

#### *Settings:*

Item	Description
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.

#### *Discussion:*

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.

### **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. This feature may also not work for lenses with multi-configuration data that alters the thicknesses or glass types of any surfaces from the fold surface onward.

### **Delete Fold Mirror**

#### *Purpose:*

Deletes an existing fold mirror, including any neighboring coordinate breaks.

#### *Settings:*

Item	Description
Fold surface	Selects which mirror surface will be deleted.

#### *Discussion:*

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” on page 207 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.

## **Export Data**

### **Export IGES/SAT/STEP Solid**

#### *Purpose:*

Exports the current lens data to a file as a collection of 3D solids in IGES, SAT, or STEP format, with various options. See also the Export IGES Line Work feature.

#### *Settings:*

Item	Description
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".
Split NSC Rays, 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 107.
Ray Pattern	Selects the type of ray set to export. This control is very similar to that defined for the 3D layout plot; see “3D Layout” on page 100. The solid beam option exports a solid volume representing the envelope of the rays. The solid beam option only works properly if the marginal rays are all traceable and unvignetted and if these rays do not form a caustic at any surface. 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.
Config	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.
Delete Vignetted	If checked, rays are not included in the data export if they will be vignetted by any surface.
Surfaces as Solids	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.



Item	Description
Export Dummy Surfaces	If checked, dummy surfaces will be exported.
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 affect.
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.

### *Discussion:*

For information on the accuracy of exported surfaces see "Comments about imported objects" on page 314.

The Initial Graphics Exchange Specification (IGES) is an American National Standard whose intended purpose is to facilitate transfer of data between CAD programs. ZEMAX 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.

ZEMAX 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.

**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.

ZEMAX 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 which is described in "Global Coordinate Reference Surface" on page 93.

If exporting surfaces as solids, ZEMAX will export mirror and 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.

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 number of facets used for rendering these objects will also increase the resolution of the exported representation.

There is always the possibility that the conversion from the ZEMAX representation of surfaces and objects to the CAD file format representation will cause a loss of precision. For more information, see "Ray tracing accuracy for imported objects" on page 314.

This feature will prompt for the file name and path to save the exported data to. The default file is EXPORT.IGS in the current output directory. If the file name provided ends in the extension IGS; then the output file will be in IGES format. If the file name extension is SAT, the output will be in SAT format. If the file ends in the STP extension, the output will be in STEP format. Otherwise, the output will be in IGES format.

To import IGES, SAT, and STEP files see “Imported” on page 314.

## **Export IGES Line Work**

### ***Purpose:***

Exports the current lens data as an IGES line work format file, with various options. See also the Export IGES/STEP Solid feature.

### ***Settings:***

Item	Description
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; see “3D Layout” on page 100 for details.
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.

### ***Discussion:***

For information on the accuracy of exported surfaces see “Comments about imported objects” on page 314.

The Initial Graphics Exchange Specification (IGES) is an American National Standard whose intended purpose is to facilitate transfer of data between CAD programs. ZEMAX currently supports version 5.2 of the IGES standard. For more information on IGES, contact U.S. Product Data Association, P. O. Box 3310, Gaithersburg, MD 20885-3310.

ZEMAX exports lines, spherical arcs, and splines to represent the shape and position of each surface. ZEMAX does not export the "edge" of the lens, and makes no distinction between dummy surfaces, glass surfaces, and mirrors. The type and number of each IGES entity exported for a given surface depends upon the surface aperture, if any, and the symmetry, if any of the surface. The following table lists how ZEMAX determines the best representation of the surface. ZEMAX exports all surfaces and rays in a 3D coordinate system referenced to the global coordinate reference surface which is described in the System Menu chapter.

This feature will prompt for the file name and path to save the exported data to. The default file is EXPORT.IGS in the current output directory. The output will be in IGES format no matter what extension is chosen.

ZEMAX makes no attempt to export objects defined as part of a Non-Sequential Components surface. See the Export IGES/STEP Solid feature if this capability is required.

## IGES ENTITIES EXPORTED BY SURFACE APERTURE AND SHAPE

Surface Aperture	Surface Shape	Entities Used
None, Circular Aperture or Obscuration	Plano	One line along the X axis, one line along the Y axis, one circle in the XY plane at the semi-diameter radius.
	Spherical	One arc in the XZ plane, one arc in the YZ plane, one circle in the XY plane at the semi-diameter radius.
	Other	One spline in the XZ plane, one spline in the YZ plane, one circle in the XY plane at the semi-diameter radius.
Rectangular Aperture or Obscuration	Plano	Three lines parallel to the X axis, three lines parallel to the Y axis, arranged at the edges and center of the surface to form a grid.
	Other	Three splines parallel to the X axis, three splines parallel to the Y axis, arranged at the edges and center of the surface to form a grid when projected upon the XY plane.
Elliptical Aperture or Obscuration	Plano	One line along the X axis, one line along the Y axis, one spline defining the edge of the surface in the XY plane.
	Other	One spline along the X axis, one spline along the Y axis, one spline defining the edge of the surface.
User Defined Aperture or Obscuration	Any	One line for each user defined segment.

The line entity is IGES entity 110. The circular arc entity is IGES entity 100. The spline entity is IGES entity 112. Rays are exported as line entities. Within GRIN media, rays are exported as a series of line entities. A warning will be issued if not all surfaces within the specified range were exported.

For surfaces where both an obscuration and another shape need to be exported, use a dummy surface collocated with the surface that has the second aperture on it. For example, to export a rectangular lens with a rectangular obscuration on the surface, use two surfaces with a zero thickness between them; one with the aperture and one with the obscuration. Make sure the surfaces have the same shape by setting the surface types the same and using pick up solves where appropriate.

### Export 2D DXF File

#### *Purpose:*

Exports the current lens data as a 2D DXD format file.

#### *Settings:*

Item	Description
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.

Item	Description
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.

*Discussion:*

This feature will prompt for the file name and path to save the exported data to. The default file is EXPORT.DXF in the current output directory. The output will be in DXF format no matter what extension is chosen. This feature is only available for rotationally symmetric lenses.

## **Miscellaneous**

### **Reverse Elements**

*Purpose:*

Reverses a lens element or group.

*Settings:*

Item	Description
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.

*Discussion:*

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. Check the reversed system results carefully to verify that the desired result was achieved.

### **Tilt/Decenter Elements**

*Purpose:*

Tilts or decenters a single surface or a group of surfaces forming an element.

*Settings:*

Item	Description
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.

*Discussion:*

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" on page 239.

## **Scale Lens**

*Purpose:*

Scale 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:*

Item	Description
Scale by factor	If selected, then a scale factor may be entered directly.
Scale by units	If selected, then the lens will be converted by the selected units.

*Discussion:*

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, the Extra 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 operand PAR3, that modify general parameter data of surfaces or objects. These data items should be scaled manually, if required.

Data items that are not scaled properly should be reported to ZEMAX technical support.

**Make Focal**

*Purpose:*

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.

**Quick Focus**

*Purpose:*

Quickly focuses an optical system by adjusting the back focal distance.

*Settings:*

Item	Description
Spot Size Radial	Focus to best RMS spot radius image plane.
Spot Size X Only	Focus to best RMS x direction spot size image plane.
Spot Size Y Only	Focus to best RMS y direction spot size image plane.
Wavefront Error	Focus to best RMS wavefront error image plane.
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 plane. The thickness chosen is that which minimizes the RMS aberration. There are several different RMS computations that may be performed as described in the "options" table 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.

## Convert to NSC Group

### *Purpose:*

Converts a range of surfaces in the Lens Data Editor into a group of components in a Non-Sequential Components surface.

### *Settings:*

Item	Description
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.

### *Discussion:*

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 ZEMAX to convert sequential surfaces to NSC objects:

The surfaces must all be of type Standard, Even Aspheric, Toroidal, or Coordinate Break. This restriction will be relaxed as more NSC object types and more conversion logic is added to ZEMAX.

All coordinate break surfaces must be between elements, in "air" and not imbedded within the lenses.

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.

ZEMAX 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 ZEMAX automatically creates. For more information on UDA's see "User defined apertures and obscurations" on page 69.

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 will 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, ZEMAX 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 standard surfaces (for mirrors) or NSC standard or toroidal lenses (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.

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.

The most important point: Don't assume this feature perfectly converts your sequential surface data; carefully check the conversion results before doing any important analysis.

### Replicate Object

#### *Purpose:*

Replicates an existing object into an array of identical objects.

#### *Settings:*

Item	Description
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 pick up 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" on page 345.

#### *Discussion:*

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.

### Ghost Focus Generator

#### *Purpose:*

Ghost focus analysis. For a description of ghosts, see "Ghost reflections" on page 48.

#### *Settings:*

Item	Description
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 disk.

Item	Description
Image Plane 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" on page 183.

*Discussion:*



***This feature may not work correctly for systems which contain coordinate breaks, multi-configuration data, or some special surface types such as the non-sequential components surface.***

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 "NON-SEQUENTIAL COMPONENTS" on page 289.***

For each ghost system, the marginal ray height, paraxial ray F/#, and 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 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 GHffss.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.

This feature may not work correctly for systems that have coordinate breaks, or if the lens is under multi-configuration control. This latter problem is circumvented by activating the configuration of interest, and then deleting all multi-configuration data before executing this feature. Not all ghost reflection configurations can be traced; there are occasionally problems with total internal reflection or rays completely missing surfaces. The GHffss.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.

## Performance Test

### *Purpose:*

System ray trace and update performance test.

### *Discussion:*

Performance test runs a check on the number of ray-surfaces per second and the number of system updates per second the computer hardware/lens combination is capable of.

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.

## Lock/Unlock All Windows

### *Purpose:*

Locks or Unlocks all open analysis windows at once.

### *Discussion:*

This tool locks or unlocks all windows. For a description of the lock function, see “Graphic windows operations” on page 36.

## Create Polygon Object

### *Purpose:*

Creates POB files of basic geometrical shapes which may be used as NSC detector objects.

### *Discussion:*

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 directory, ready to use by the NSC editor.



***Suggestions for additional basic POB shapes should be directed to technical support.***

For information on POB objects, see “Polygon Object” on page 319. For information on using objects as detectors, see “Objects as detectors” on page 335. For information on the POB format, see “Defining Polygon Objects” on page 373.

## Slider

### *Purpose:*

The slider control is used to interactively adjust any system or surface parameter while viewing any analysis window.

### *Settings:*

Item	Description
The data to be modified is defined by the following three settings:	
Group (item name not shown)	Select either the surface, system, configuration, or NSC data group.
Data (item name not shown)	Select the data 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 multi-configuration operands are listed. For NSC data, object position and parameter data are listed.
Number (item name not shown)	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	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.
Save	Saves the current data, but leaves the slider control box open.
Exit	Restores the original (or last saved) 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.

## **Introduction**

This chapter provides detailed descriptions of each of the report features ZEMAX supports. The Settings menu option allows the default parameters for the calculation to be changed. Selecting this option will cause a feature dialog box to appear on the screen. The dialog box typically has five buttons: OK, Cancel, Save, Load, and Reset. OK causes the window to recalculate and redisplay the data with the currently selected options. Cancel reverts the options to those selected before the dialog box appeared, and does not update the data in the window. Save stores the currently selected options as defaults for future use, and then causes the window to recalculate and redisplay the data. Load loads the most recently saved default options, but does not exit the dialog box. Reset returns the options to the “factory default” values, but does not exit the dialog box. A report window can be updated by double clicking with the left mouse button within the window. The settings dialog box is invoked by clicking with the right mouse button.

## **Surface Data**

### *Purpose:*

Display surface data.

### *Settings:*

Item	Description
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 in 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” on page 485.

## **System Data**

### *Purpose:*

Display system data.

### *Settings:*

None.

### *Discussion:*

This feature generates a text window which lists many system-related parameters, such as pupil positions and sizes, magnification, F/#, etc.

## **Prescription Data**

### *Purpose:*

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.

*Settings:*

Item	Description
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” on page 93 for a discussion of this data.
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.

*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 are 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 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.

**Report Graphics 4/6**

*Purpose:*

This function generates a graphical window that simultaneously displays either 4 or 6 analysis graphics. The primary advantage of this feature is that multiple analysis graphs may be printed on a single page, making a suitable summary for reports, archival documentation, or promotional literature.

*Settings:*

The report graphics window works somewhat differently from other analysis windows. If the “Settings” option is selected from the window's menu bar, then a dialog box will appear which allows selection of the type of graphic

to be displayed in each position within the window. The selected graphs to display can be saved as default selections like any other window. The configuration number for each window may also be selected.

Additionally, the settings may be saved as a custom report for later re-use. The Save As New Report button will save the current settings in a file. Once the settings are saved, the name will appear as a menu option for quick regeneration of the plot. To change the settings for an individual graph within the window, the right mouse button must be used. First, unzoom the display (if the graph has been zoomed), then right mouse click anywhere in the window within the region of the plot whose settings need changing.



### **Edit/Run ZPL Macros**

#### *Purpose:*

Runs ZEMAX Programming Language macros. This option invokes a dialog box which allows editing, viewing, and executing of macros.

#### *Discussion:*

See the chapter “ZEMAX Programming Language” for details on the ZPL macro language.

This dialog box is most useful for developing and debugging new macros. It also displays a terminate button, which allows the macro execution to be terminated.

### **Refresh Macro List**

#### *Purpose:*

Updates the macro list.

#### *Discussion:*

This feature updates the macro list; which may be required if any macros have been added or deleted since the last time the list was refreshed.

### **Macro Names**

#### *Purpose:*

A list of all ZPL macros in the default directory for macros is shown. By clicking on the macro name, it immediately executes.

#### *Discussion:*

It is much faster to execute a macro from the list of macro names than to use the ZPL Macros dialog box. To terminate a ZPL macro, press the escape key.

### **Extensions**



***This feature is only available in the EE edition of ZEMAX.***

---

#### *Purpose:*

Runs ZEMAX extensions.

#### *Discussion:*

See “ZEMAX EXTENSIONS” on page 607 for details on creating ZEMAX extensions. The extensions supplied with ZEMAX are described in the following table.

EXTENSIONS INCLUDED WITH ZEMAX

Extension Name	Description
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.

Extension Name	Description
SagCalculation	Creates an extended listing of sag data for non-radial symmetric surfaces.
TransmissionPlot	Computes transmission vs. Wavelength.

## **Refresh Extensions List**

### *Purpose:*

Updates the list of extensions.

### *Discussion:*

This feature updates the list of extensions that appears in the Extensions menu; 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 \Extend subdirectory of the main ZEMAX directory.

## **Extension Names**

### *Purpose:*

A list of all ZEMAX extensions in the \ZEMAX\Extend directory is shown. Clicking on the extension name will execute that extension.

### *Discussion:*

See “ZEMAX EXTENSIONS” on page 607 for details on creating and executing ZEMAX extensions.

## **Listing of open windows**

### *Purpose:*

This menu lists all currently open windows by window title. Clicking on any window title listed will bring that window to the front of the display and give that window the input focus.

### *Discussion:*

Clicking on any window title listed will bring that window to the front of the display and give that window the input focus.



## **Introduction**

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**

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" on page 65.

## **Extra data**

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 would make for a very large spreadsheet, and so a separate editor is used for the extra data values. However, the concept is the same. The extra data values are shared by each surface type, and change meaning depending upon the surface type selected. The column headings in the Extra Data Editor also change as the cursor is moved from surface to surface.

## **Summary of 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 screen 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 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 on page "User Defined" on page 275. 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.

### Surface types

The types of surfaces supported by ZEMAX are summarized in the following table. The most common surface type is the Standard surface described on page 270. For types not listed, see “User Defined” on page 275.

#### SUMMARY OF SURFACE TYPES

Surface Model	Description	Page
ABCD	Uses ABCD matrix to model a “black box”.	228
Alternate Even	Even Asphere surface with alternate solution selected.	228
Alternate Odd	Odd Asphere surface with alternate solution selected.	229
Atmospheric Refraction	Refraction caused by looking through Earth's atmosphere.	229
Biconic	A conic asphere in X and Y independently.	229
Biconic Zernike	A biconic surface with x, y, and Zernike polynomial terms added.	230
Binary Optic 1	Uses 230 term polynomial to define phase.	230
Binary Optic 2	Uses radial polynomial to define phase.	232
Binary Optic 3	Dual zone aspheric and diffractive surface.	
Birefringent In/Out	For modeling uniaxial crystals; supports tracing of rays in ordinary or extraordinary mode.	235
Conjugate	Defines surface with perfect imaging at two points.	238
Coordinate Break	Permits rotation and decentration.	239
Cubic Spline	Rotationally symmetric fit to 8 points.	240
Cylindrical Fresnel	Polynomial cylindrical Fresnel on a polynomial cylindrical surface.	240
Diffraction Grating	Ruled grating on standard surface.	241
Elliptical Grating 1	Elliptical grating with aspheric terms and polynomial grooves.	242
Elliptical Grating 2	Elliptical grating with aspheric terms and grooves formed by tilted planes.	243
Even Asphere	Standard surface plus polynomial asphere terms.	243
Extended Asphere	Uses radial polynomial to define sag.	244
Extended Cubic Spline	Rotationally symmetric fit of up to 250 points.	245
Extended Fresnel	Polynomial Fresnel on a polynomial surface.	245
Extended Odd Asphere	Uses odd terms of radial powers.	246
Extended Polynomial	Uses 230 term polynomial expansion to define sag.	247
Extended Toroidal Grating	An aspheric toroidal grating with extended polynomial terms	248
Fresnel	Plane surface with refractive power.	249
Generalized Fresnel	XY polynomial Fresnel on an aspheric substrate.	249

Surface Model	Description	Page
Gradient 1	Radial gradient index material surface.	250
Gradient 2	Radial gradient index material surface.	251
Gradient 3	Radial and axial gradient index material surface.	251
Gradient 4	X, Y, and Z gradient index material surface.	252
Gradient 5	Radial and axial gradient index material surface with dispersion model.	252
Gradient 6	Radial gradient index material surface with Gradient Lens Corp. dispersion model.	253
Gradient 7	Spherical gradient profile.	254
Gradium™	Axial gradient index material surface with dispersion model.	255
Gradient 9	Radial gradient index material surface with NSG SELFOC lens dispersion model.	257
Gradient 10	Y gradient index material surface with dispersion model.	258
Grid Phase	A phase surface described by a grid of points.	258
Grid Sag	A surface shape described by a grid of points.	259
Hologram 1	Two-point optically fabricated hologram.	261
Hologram 2	Two-point optically fabricated hologram.	262
Irregular	A standard surface with decenter, tilt and other deformations.	262
Jones Matrix	General Jones Matrix for modifying polarization state.	263
Non-Sequential Components	For tracing rays non-sequentially through a collection of 3D surfaces and objects.	275
Odd Asphere	Standard surface plus polynomial asphere terms.	263
Odd Cosine	Odd Asphere surface plus cosine polynomial terms.	263
Optically Fabricated Hologram	Optically fabricated hologram with arbitrary construction optics and elliptical substrate.	264
Paraxial	Thin lens surface, has ideal behavior.	266
Paraxial XY	Thin lens with separate specification in X,Y.	267
Periodic	Cosine shaped surface.	267
Polynomial	Polynomial expansion in x and y.	268
Radial Grating	A diffraction grating with radial phase profile	268
Radial NURBS	Uses NURBS curve to define a rotationally symmetric surface.	269
Retro Reflect	Retro reflects rays back along incident path.	240
Standard	Includes planes, spheres, and conics.	270
Superconic	Superconic asphere with fast convergence.	271
Tilted	Defines a tilted surface without changing coordinate systems.	239

Surface Model	Description	Page
Toroidal	Conic and aspheric toroids and cylinders.	272
Toroidal Grating	Ruled grating on a conic toroid.	240
Toroidal Hologram	Toroidal substrate with two point optically fabricated hologram.	273
Toroidal NURBS	Uses NURBS curve to define a toroidally symmetric surface.	274
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.	275
Variable Line Space Grating	Variable line space grating surface.	282
Zernike Fringe Phase	Uses 37 Zernike Fringe polynomials to define phase.	282
Zernike Fringe Sag	Uses 37 Zernike Fringe polynomials to define sag.	283
Zernike Standard Phase	Uses up to 231 Zernike Standard polynomials to define surface phase.	284
Zernike Standard Sag	Uses up to 231 Zernike Standard polynomials to define surface sag.	285
Zone Plate	Fresnel Zone Plate model using annular rings of varying depth.	286

## **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, considering the Grid Phase surface with an externally provided data file. For information on the Grid Phase surface, see "Grid Phase" on page 258.

### **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:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x^2 x^2 - (1 + k_y)c_y^2 y^2}},$$

where

$$c_x = \frac{1}{R_x}, c_y = \frac{1}{R_y}.$$

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:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - (1 + k_x)c_x^2 x^2 - (1 + k_y)c_y^2 y^2}} + \sum_{i=1}^{16} \alpha_i x^i + \sum_{i=1}^{16} \beta_i y^i + \sum_{i=1}^N A_i Z_i(\rho, \phi),$$

where

$$c_x = \frac{1}{R_x}, c_y = \frac{1}{R_y},$$

and the  $Z_i$  terms are the Zernike standard terms as described in “Zernike Standard Sag” on page 285. 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

Extra Data Number	Description
1-16	Coefficients on powers of $x$
17-32	Coefficients on powers of $y$
33-242	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" on page 243 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,  $A_i$  is the coefficient on the  $i^{th}$  extended polynomial term, and  $M$  is the diffraction order. The polynomials are a power series in  $x$  and  $y$ , as described in "Extended Polynomial" on page 247. The coefficients  $A_i$  all have units of radians ( $2\pi$  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	$\alpha_1 - \alpha_8$	Absolute? (see discussion)

#### EXTRA DATA DEFINITIONS FOR BINARY OPTIC 1 SURFACES

Extra Data Number	Description
1	Maximum term number.
2	Normalization radius. All ray-intercept points are divided by this number to determine the $x$ and $y$ coordinates for polynomial evaluation.
3-232	Polynomial terms.
233 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, and NOT 15, which is the "extra data number", which is the position in the extra data list of parameters. The "extra data number" is always two greater than the term number!



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" on page 258.

### 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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16},$$

where the terms are identical to those in the Even Asphere Surface model. See "Even Asphere" on page 243 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  $2i^{th}$  power of  $\rho$ , 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  $\rho^{480}$  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	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$



## EXTRA DATA DEFINITIONS FOR BINARY OPTIC 2 SURFACES

Extra Data Number	Description
1	Maximum term number.
2	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation.
3	Coefficient in radians on $\rho^2$
4	Coefficient in radians on $\rho^4$
n+2	Coefficient in radians on $\rho^{2n}$

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  $\rho^{14}$ , 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, and NOT 9, which is the "extra data number", which is the position in the extra data list of parameters. The "extra data number" is always two greater than the term number!



***The "extra data number" is always two greater than the term number!***

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" on page 258.

### Binary optic coefficients sign conventions

See "Binary Optic 1" on page 230 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. 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,  $\beta_{1i}$  is the coefficient on the  $2i^{th}$  power of  $\rho_1$ , which is the normalized radial aperture coordinate, and  $M_1$  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  $\delta_o$  is temporarily set to zero while evaluating  $\Phi_2$  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  $\delta_o$  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.

## PARAMETER DEFINITIONS FOR BINARY OPTIC 3 SURFACES

Param 1	Param 2	Param 3	Param 4	Param 5	Param 6	Param 7
$R_2$	$k_2$	$A_1$	$A_2$	$M_1$	$M_2$	$\sin \delta_o$

## EXTRA DATA DEFINITIONS FOR BINARY OPTIC 3 SURFACES

Extra Data Number	Description
1	Maximum term number, N.
2, 3, 4, 5	The coefficients $\alpha_{11}$ , $\beta_{11}$ , $\alpha_{21}$ , and $\beta_{21}$ if $N \geq 1$ .
6, 7, 8, 9	The coefficients $\alpha_{12}$ , $\beta_{12}$ , $\alpha_{22}$ , and $\beta_{22}$ if $N \geq 2$ .
10, 11, 12, 13, etc.	The pattern continues in groups of 4 coefficients for N groups.

### Binary optic coefficients sign conventions

See "Binary Optic 1" on page 230 for a discussion of sign conventions.

### **Birefringent In and Birefringent Out**



***This feature is only available in the EE edition 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. For information on ray tracing details, see Quan-Ting Liang, "Simple ray tracing formulas for uniaxial optical crystals", *Applied Optics* Vol. **29**, No. 7, (1990). 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" on page 263.

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  $\theta_w$ , which is the angle between the crystal axis vector  $\hat{a}$  and the refracted wave vector  $\hat{k}$ . Additionally, the ray vector  $\hat{s}$ , which is the vector pointing in the direction of energy propagation (also called the Poynting vector), 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  $\theta_w$  is defined by

$$\cos \theta_w = \hat{k} \cdot \hat{a}.$$

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  $\alpha$  between  $\hat{k}$  and  $\hat{s}$  is defined by

$$\cos \alpha = \hat{k} \cdot \hat{s}, \text{ where}$$

$$\tan \alpha = \frac{(n_e^2 - n_o^2) \tan \theta_w}{n_e^2 + (n_o \tan \theta_w)^2},$$

and the vectors  $\hat{k}$  and  $\hat{s}$  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.

Because the birefringence effects both the refraction into the media, as well as the refraction out of the media, two surfaces are required to model a birefringent element: the "birefringent in" and the "birefringent out" surfaces. Each "in" must be followed by an "out" surface; if this is violated ZEMAX will issue an error message and not trace any rays. The only exception is that any number of coordinate break surfaces (or surface tilts and decenters) may be placed between the "in" and "out" surfaces. The ray is propagated to the next surface using the  $\hat{s}$  vector components; however the refraction out of the birefringent media is governed again by the  $\hat{k}$  vector components.

### 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 orientation

The crystal axis orientation is defined using parameter columns 2, 3, and 4 for the x-, y-, and z- direction cosines of the crystal axis ray. 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 surface; and ZEMAX converts to unit vector components.

### 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. 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. 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.

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.

### Transmission and other properties of birefringent surfaces

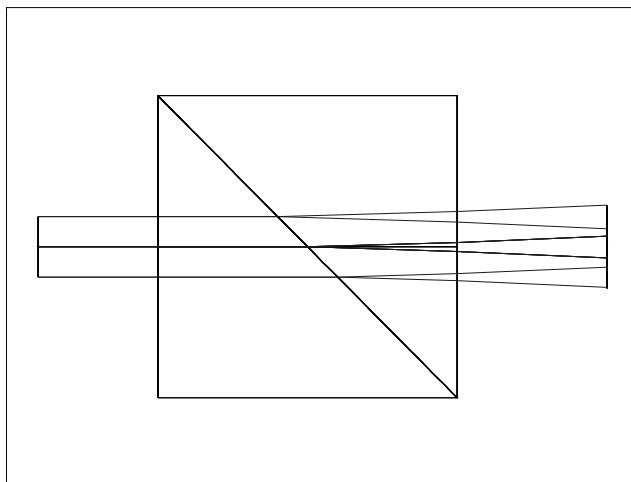
When light propagates through birefringent media, the index of refraction of the glass is different for the S and P polarizations. The ordinary index is what is seen by the perpendicular, or S-polarized light, while the effective index is seen by the parallel or P-polarized light. Note the S and P polarization 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.

### Sample ZEMAX files

For some samples of the birefringent surface type, see the "\\Samples\\Sequential\\Birefringent prisms" directory. 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	Parameter 2	Parameter 3	Parameter 4	Parameter 5
Length to draw crystal axis, 0 to not draw.	Mode, 0, 1, 2, or 3.	X-cosine	Y-cosine	Z-cosine	Parax Ignore. If anything other than 0, paraxial rays ignore the extraordinary index.

The birefringent "out" surface does not use any parameters. The surface shape of the birefringent in and out surfaces is the same as a standard surface; which includes planes, spheres, and conic aspheres; all symmetric about the local Z axis.

### **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  $z_1$  and  $z_2$  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  $z_1$  and  $z_2$  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  $z_1$  and  $z_2$  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  $z_1$  nor the  $z_2$  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" on page 73 for details.



***The order of the decenters and tilts matters!***

If the "order" flag is set to zero, ZEMAX first decenters in  $x$ , and then in  $y$  (because these coordinates are orthogonal which is first does not matter). Then ZEMAX tilts about the local  $z$  axis, then the local  $y$  axis, then the local  $x$  axis. Although the order of rotations seems reversed from "normal" this convention is used because it is equivalent to rotating around the  $x$  axis, then rotating around the "new"  $y$  axis resulting from the  $x$  rotation; then finally rotating around the new  $z$  axis which resulted from the previous  $x$  and  $y$  rotations. Either perspective is valid.

If the "order" flag is any other value (such as unity), then the tilts are done first, in the order  $x$ ,  $y$ , and  $z$ , and then 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.

The coordinate rotations are described by a series of three rotation matrices. If the "order" flag is not zero the coordinate rotation is described by:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(\theta_z) & -\sin(\theta_z) & 0 \\ \sin(\theta_z) & \cos(\theta_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta_y) & 0 & \sin(\theta_y) \\ 0 & 1 & 0 \\ -\sin(\theta_y) & 0 & \cos(\theta_y) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_x) & -\sin(\theta_x) \\ 0 & \sin(\theta_x) & \cos(\theta_x) \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

If the order flag is zero, then the z and x matrices are exchanged, so the z rotation is first, followed by y, then x.

## 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" on page 245. For a more general non-rotationally symmetric surface, see "Grid Sag" on page 259.

### **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 (a) the curve goes through the defined points, (b) the curve be continuous across segment boundaries, and (c), the first derivatives are also continuous across the segment boundaries. For a third order spline, it is not possible to require higher order derivatives, such as second or third, to be continuous across segment boundaries. For this reason, splines are of limited accuracy and usefulness in high precision optical design. Note that fundamental optical properties, such as surface power, are determined by second order derivatives, and basic aberrations such as coma and spherical are controlled by third and fourth order derivatives.

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" on page 243. 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 in the Extra Data Editor. However, the Extra Data Editor 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 CYLINDER FRESNEL SURFACES

Extra Data Number	Description
1	Maximum aspheric polynomial term number, n. The maximum is 8.
2	The Y direction curvature (NOT RADIUS) of the Fresnel surface. This parameter will effect the refraction of the surface, but not the shape of the substrate.
3	The Y direction conic constant of the Fresnel surface. This parameter will effect the refraction of the surface, but not the shape of the substrate.
4	Coefficient on $y^2$
n+3	Coefficient on $y^{2n}$

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.

### **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),  $\theta_2$  is the refracted angle,  $\theta_1$  is the incident angle, M is the diffraction order,  $\lambda$  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),  $\theta_2$  is the refracted angle,  $\theta_1$  is the incident angle, M is the diffraction order,  $\lambda$  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:

$$\frac{1}{T} = d_{eff} = \frac{1}{T_0} + \alpha y + \beta y^2 + \Gamma y^3.$$

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" on page 247.

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	Parameter 6	Parameter 7	Parameter 8
Lines per micrometer, $T_0$	Diffraction order	a	b	c	$\alpha$	$\beta$	$\Gamma$

### **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  $\theta_y$ , 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" on page 242.

#### 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	$\theta_y$

### **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
$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$

### **Extended Asphere**

The extended asphere surface is similar to the Even Asphere surface; see “Even Asphere” on page 243. 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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^N \alpha_i \rho^{2i},$$

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  $\rho$  is used because then the coefficients  $\alpha_i$  all have units of lens units. The extra data terms are defined in the following table.

### EXTRA DATA DEFINITIONS FOR EXTENDED ASPHERE SURFACES

Extra Data Number	Description
1	Maximum term number. The maximum is 240.
2	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation.
3	Coefficient in lens units on $\rho^2$
4	Coefficient in lens units on $\rho^4$
n+2	Coefficient in lens units on $\rho^{2n}$

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  $\rho^{14}$ , 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, and NOT 9, which is the "extra data number", which is the position in the extra data list of parameters. The "extra data number" is always two greater than the term number!



***The "extra data number" is always two greater than the term number!***

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” on page 259 or “User Defined” on page 275.

## **Extended Cubic Spline**

The Extended Cubic Spline surface is similar to the Cubic Spline surface (see “Cubic Spline” on page 240), 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” on page 240. For a non-rotationally symmetric spline surface, see “Grid Sag” on page 259.

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 load capability of the extra data editor. This feature is described in “Extra Data” on page 79.

### **EXTRA DATA DEFINITIONS FOR EXT. CUBIC SPLINE SURFACES**

Extra Data Number	Description
1	Maximum number of spline points.
2	Step size in lens units between points along the radius.
3	The sag at a distance of one step size away from the vertex.
4...n	The sag at a distance of 2... (n-2) 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” on page 259.

## **Extended Fresnel**

The Fresnel surface described in “Fresnel” on page 249 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” on page 243 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  $Z_f$  (below) and the substrate shape expression for  $Z_s$  (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  $\alpha$ ) 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 in the Extra Data Editor. However, the Extra Data Editor 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

Extra Data Number	Description
1	Maximum aspheric polynomial term number, n. The maximum is 8.
2	The curvature (NOT RADIUS) of the Fresnel surface. This parameter will effect the refraction of the surface, but not the shape of the substrate.
3	The conic constant of the Fresnel surface. This parameter will effect the refraction of the surface, but not the shape of the substrate.
4	Coefficient on $r^2$
n+3	Coefficient on $r^{2n}$

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" on page 263. 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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^N \alpha_i \rho^i,$$

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  $\rho$  is used because then the coefficients  $\alpha_i$  all have units of lens units. The extra data terms are defined in the following table.

#### EXTRA DATA DEFINITIONS FOR EXTENDED ODD ASPHERE SURFACES

Extra Data Number	Description
1	Maximum term number. The maximum is 240.
2	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation.
3	Coefficient in lens units on $\rho^1$

Extra Data Number	Description
4	Coefficient in lens units on $\rho^2$
n+2	Coefficient in lens units on $\rho^n$

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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \sum_{i=1}^N A_i E_i(x, y).$$

where  $N$  is the number of polynomial coefficients in the series, and  $A_i$  is the coefficient on the  $i^{th}$  extended polynomial term. The polynomials are a power series in  $x$  and  $y$ . The first term is  $x$ , then  $y$ , then  $x^2$ ,  $x^2y$ ,  $y^2$ , 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  $x^2y^2$ . The coefficients  $A_i$  all have units which are the same as the lens units, such as millimeters or inches.

#### EXTRA DATA DEFINITIONS FOR EXT. POLYNOMIAL SURFACES

Extra Data Number	Description
1	Maximum term number.
2	Normalization radius. All ray-intercept points are divided by this number to determine the $x$ and $y$ coordinates for polynomial evaluation.
3-232	Polynomial terms.
233 and above	Not used.

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, and NOT 15, which is the "extra data number", which is the position in the extra data list of parameters. The "extra data number" is always two greater than the term number! The maximum term number is the same as " $N$ " in the surface sag equation.



***The "extra data number" is always two greater than the term number!***

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 (page 272), toroidal grating (page 240), and extended polynomial (page 247) 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:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14} + \alpha_8 y^{16}.$$

This curve is similar to the even asphere (page 243) 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  $i^{th}$  extended polynomial term. The polynomials are a power series in  $x$  and  $y$ . The first term is  $x$ , then  $y$ , then  $x^2$ ,  $x^2y$ ,  $y^2$ , 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  $x^2y^2$ . 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**

Extra Data Number	Description
1	Maximum term number.
2	Normalization radius. All ray-intercept points are divided by this number to determine the $x$ and $y$ coordinates for polynomial evaluation.



Extra Data Number	Description
3-232	Polynomial terms.
233 and above	Not used.

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, and NOT 15, which is the "extra data number", which is the position in the extra data list of parameters. The "extra data number" is always two greater than the term number! The maximum term number is the same as "N" in the surface sag equation. 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.

## **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 spherical (or aspherical) 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.

### PARAMETER DEFINITIONS FOR FRESNEL SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$

## **Generalized Fresnel**

This surface represents a generalization of the Fresnel surface described in "Fresnel" on page 249. 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" on page 243. 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 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  $i^{th}$  extended polynomial term. The polynomial terms are identical to those described in "Extended Polynomial" on page 247. 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
$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$

#### EXTRA DATA DEFINITIONS FOR EXT. POLYNOMIAL SURFACES

Extra Data Number	Description
1	Maximum term number.
2	Normalization radius. All ray-intercept points are divided by this number to determine the x and y coordinates for polynomial evaluation.
3-232	Polynomial terms.
233 and above	Not used.

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**

Media whose index of refraction is described by

$$n = n_0 + n_{r2}r^2 + n_{r1}r,$$

where  $r^2 = x^2 + y^2$ , can be modeled using the gradient 1 surface. Four parameters are required: the maximum step size,  $\Delta t$ , 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  $\Delta t$  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
$\Delta t$	$n_0$	$n_{r2}$

### **Gradient 2**

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  $r^2 = x^2 + y^2$ , can be modeled using the gradient 2 surface. Eight parameters are required: the maximum step size,  $\Delta t$ , 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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see “Discussion on maximum step size for GRIN surfaces” on page 250. Also see “Restrictions on surfaces following GRIN surfaces” on page 251.

#### PARAMETER DEFINITIONS FOR GRADIENT 2 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\Delta t$	$n_0$	$n_{r2}$	$n_{r4}$	$n_{r6}$	$n_{r8}$	$n_{r10}$	$n_{r12}$

### **Gradient 3**

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  $r^2 = x^2 + y^2$ , can be modeled using the gradient 3 surface. Eight parameters are required: the maximum step size,  $\Delta t$ , the base index,  $n_0$ , and the remaining six coefficients of the preceding equation. Note that some of the coefficients have units.

The maximum step size  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see “Discussion on maximum step size for GRIN surfaces” on page 250. Also see “Restrictions on surfaces following GRIN surfaces” on page 251.

## PARAMETER DEFINITIONS FOR GRADIENT 3 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\Delta t$	$n_0$	$n_{r2}$	$n_{r4}$	$n_{r6}$	$n_{z1}$	$n_{z2}$	$n_{z3}$

### **Gradient 4**

Media whose index of refraction is described by

$$n = n_0 + n_{x1}x + n_{x2}x^2 + n_{y1}y + n_{y2}y^2 + n_{z1}z + n_{z2}z^2,$$

can be modeled using the gradient 4 surface model. There are eight parameters required: the maximum step size,  $\Delta t$ , the base index,  $n_0$ , 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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces" on page 250. Also see "Restrictions on surfaces following GRIN surfaces" on page 251.

## PARAMETER DEFINITIONS FOR GRADIENT 4 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\Delta t$	$n_0$	$n_{x1}$	$n_{x2}$	$n_{y1}$	$n_{y2}$	$n_{z1}$	$n_{z2}$

### **Gradient 5**

The gradient 5 surface has the gradient form:

$$n_{ref} = n_0 + n_{r2}r^2 + n_{r4}r^4 + n_{z1}z + n_{z2}z^2 + n_{z3}z^3 + n_{z4}z^4,$$

where  $r^2 = x^2 + y^2$ . Eight parameters are required: the maximum step size,  $\Delta t$ , the base index,  $n_0$ , and the remaining six coefficients of the preceding equation. 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:

$$n(\lambda)^2 = n(\lambda_{ref})^2 + \sum_{i=1}^3 \frac{K_i(\lambda^2 - \lambda_{ref}^2)}{\lambda^2 - L_i}, \text{ where}$$

$$K_i = \sum_{j=1}^{K\_MAX} K_{ij}[n_{ref}]^{j-1}, L_i = \sum_{j=1}^{L\_MAX} L_{ij}[n_{ref}]^{j-1}, \text{ and } n_{ref} = n(\lambda_{ref}).$$

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 \GLASSCAT directory.

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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces" on page 250. Also see "Restrictions on surfaces following GRIN surfaces" on page 251.

## PARAMETER DEFINITIONS FOR GRADIENT 5 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\Delta t$	$n_0$	$n_{r2}$	$n_{r4}$	$n_{z1}$	$n_{z2}$	$n_{z3}$	$n_{z4}$

## **Gradient 6**

The gradient 6 surface has the following gradient profile:

$$n = n_0 + n_1 r^2 + n_2 r^4.$$

The difference between the gradient 6 and gradient 1 surface models is that the gradient 6 surface uses dispersion formulas to compute  $n_0$ ,  $n_1$ , and  $n_2$  automatically rather than using entries from the lens data editor.

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  $n_0$ ,  $n_1$  and  $n_2$ . This expression uses nanometers rather than micrometers for  $\lambda$ .

The dispersion data is user defined and is stored in an ASCII file called GLC.DAT. The GLC.DAT file contains blocks of 13 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 10 characters long. The next 12 lines are the values for A, B, C, and D for  $n_0$ ,  $n_1$ , and then  $n_2$ , respectively. There are no blank lines allowed between blocks. 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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces" on page 250. Also see "Restrictions on surfaces following GRIN surfaces" on page 251.

## PARAMETER DEFINITIONS FOR GRADIENT 6 SURFACES

Parameter 1
$\Delta t$

## **Gradient 7**

The gradient 7 surface has a gradient profile with spherical symmetry. The index of refraction is described by

$$n = n_0 + \alpha(r - R) + \beta(r - R)^2, \text{ where}$$

$$r = \frac{R}{|R|} \sqrt{x^2 + y^2 + (R - z)^2}.$$

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,  $\Delta t$ , the base index,  $n_0$ , R,  $\alpha$ , and  $\beta$ . Note that  $\alpha$  and  $\beta$  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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see “Discussion on maximum step size for GRIN surfaces” on page 250. Also see “Restrictions on surfaces following GRIN surfaces” on page 251.

## PARAMETER DEFINITIONS FOR GRADIENT 7 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5
$\Delta t$	$n_o$	$R$	$\alpha$	$\beta$

### **GRADIUM™**

This surface type models lenses made from stock gradient index blanks available from LightPath Technologies, Inc. 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  $\Delta z$  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,  $\Delta z$ . 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 \ZEMAX\GLASSCAT directory. To select a different profile file, see “GRADIUM Profile” on page 91. For a list of the available profiles, see “Gadium™ Profile” on page 180.

### **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  $n_0$  to  $n_{11}$ . 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  $\Delta z$  to yield the specified reference index. However,  $\Delta z$  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:

$$z' = t_x(x - d_x) + t_y(y - d_y) + t_z z, \text{ where}$$

$$t_z = [1.0 - t_x^2 - t_y^2]^{\frac{1}{2}},$$

and  $t_x$ ,  $t_y$ ,  $t_z$  are the coefficients of the unit vector which points along the axial gradient axis and  $d_x$  and  $d_y$  are the decenters in lens units of the start of the profile. If  $t_x$  and  $t_y$  are both zero, then the  $d_x$  and  $d_y$  values do not matter (since the gradient is only along the  $z$  axis) and the  $t_z$  value is unity. The terms  $t_x$  and  $t_y$  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  $t_x$  and  $t_y$ .

The tolerance terms,  $d_x$ ,  $d_y$ ,  $t_x$ , and  $t_y$ , 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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces" on page 250. Also see "Restrictions on surfaces following GRIN surfaces" on page 251.

## 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	$\Delta t$	$\Delta z$	$n_{ref}$	$d_x$	$d_y$	$t_x$	$t_y$	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 "sag" or z-coordinate of the gradient 9 surface is the same as the standard surface plus a "tilt" term in both x and y directions:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + x \tan(\alpha) + y \tan(\beta),$$

where  $c$  is the curvature (the reciprocal of the radius),  $r$  is the radial coordinate in lens units,  $k$  is the conic constant, and  $\tan \alpha$  and  $\tan \beta$  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. 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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces" on page 250. Also see "Restrictions on surfaces following GRIN surfaces" on page 251.

### PARAMETER DEFINITIONS FOR GRADIENT 9 SURFACES

Parameter 1	Parameter 2	Parameter 3
$\Delta t$	$\tan \alpha$	$\tan \beta$

## **Gradient 10**

The gradient 10 surface models glass with the following gradient profile:

$$n = n_0 + n_{y1}y_a + n_{y2}y_a^2 + n_{y3}y_a^3 + n_{y4}y_a^4 + n_{y5}y_a^5 + n_{y6}y_a^6,$$

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 on page 252. 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  $\Delta t$  determines the trade-off between ray tracing speed and accuracy. For details see "Discussion on maximum step size for GRIN surfaces" on page 250. Also see "Restrictions on surfaces following GRIN surfaces" on page 251.

### PARAMETER DEFINITIONS FOR GRADIENT 10 SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\Delta t$	$n_0$	$n_{y1}$	$n_{y2}$	$n_{y3}$	$n_{y4}$	$n_{y5}$	$n_{y6}$

## **Grid Phase**

This surface is nearly identical to the Grid Sag surface (see "Grid Sag" on page 259). 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” on page 230 for a discussion of sign conventions.

### PARAMETER DEFINITIONS FOR GRID PHASE SURFACES

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 260.

### Grid Sag

The grid sag surface has a shape defined by a base plane, sphere, conic asphere, or polynomial 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

$$z_{base} = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \sum_{i=1}^8 \alpha_i r^{2i}.$$

$$Z_{ij} = Z_{base} + Z(x_i, y_j), \text{ where}$$

$$x_i = xdec + delx \cdot (i - i_c), y_j = ydec + dely \cdot (j - j_c), \text{ and}$$

$$i_c = \frac{n_x + 1}{2}, j_c = \frac{n_y + 1}{2}, 1 \leq i \leq n_x, 1 \leq j \leq n_y,$$

where 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 odd and 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.

### 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 Extra Data Editor "Import" feature (see “Extra Data” on page 79). The proper file format is as follows:

```
nx ny delx dely unitflag xdec ydec
z dz/dx dz/dy d2z/dxdy no data
.
.
.
```

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.

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  $d^2z/dxdy$ ) are zero for every point in the file, then ZEMAX will automatically estimate the derivatives using a finite difference method.

### 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" on page 240.

To use the grid sag surface, first change the surface type to grid sag. Then from the Extra Data Editor, choose Tools, Import, and specify the correct surface number and DAT file name. The data will be loaded into memory, but will not be displayed on the Extra 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 Extra Data Editor.

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 disk. One way to speed up editing is to turn off the "disk multi step" undo feature in ZEMAX. See "Preferences" on page 58.

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 imbed 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, 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
Interpolation method. Use 0 for bicubic spline, 1 for linear. See "Bicubic spline vs. linear interpolation" on page 260.	$\alpha_1 - \alpha_8$ , see "Even Asphere" on page 243

### **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  $\hat{n}$  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,  $\lambda_c$  and  $\lambda_p$  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 manual, 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.

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" on page 264.



***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
$X_1$	$Y_1$	$Z_1$	$X_2$	$Y_2$	$Z_2$	$\lambda_c$	$m$

### **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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2 r^2}} + Z_s \rho^4 + Z_a \rho_y'^2 + Z \rho^2 \rho_y',$$

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 rmax is the maximum radial aperture of the lens, defined by the semi-diameter value for the surface. The coefficients Zs, Za, and Zc 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  $\theta$  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 chapter, 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	$\theta$

## **Jones Matrix**



*This feature is only available in the EE edition of ZEMAX.*

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,  $E_x$ , and  $E_y$  are all complex numbers. See "Defining polarizing components" on page 511 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 DEFINITIONS FOR JONES MATRIX SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Ar	Ai	Br	Bi	Cr	Ci	Dr	Di

## **Non-Sequential Components**



*This feature is only available in the EE edition of ZEMAX.*

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 the chapter "Non-Sequential Components" 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

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \beta_1 r^1 + \beta_2 r^2 + \beta_3 r^3 + \beta_4 r^4 + \beta_5 r^5 + \beta_6 r^6 + \beta_7 r^7 + \beta_8 r^8.$$

Note that the coefficients have units. The coefficients are entered in the corresponding parameter columns, as shown in the following table.

### PARAMETER DEFINITIONS FOR ODD ASPHERE SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$\beta_5$	$\beta_6$	$\beta_7$	$\beta_8$

## **Odd Cosine**



*This feature is only available in the EE edition of ZEMAX.*

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 value is not an integer. The  $\beta_i$  coefficients have units which depend upon the index i, and the  $A_i$  coefficients have units of length in lens units. The angle  $\theta$  is measured in radians, and so 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

Parameters 1-16	Parameter 17	Parameter 18	Parameters 19-42
The odd aspheric coefficients $\beta_i$ .	The number of cosine terms m, must be between 0 and 6.	The normalization radius R for the coordinate s in the cosine portion of the sag formula.	The A, P, B, and C terms in the cosine portion of the sag formula.

## **Optically Fabricated Hologram**



***This feature is only available in the EE edition of ZEMAX.***

The Hologram 1, Hologram 2, and Toroidal Hologram surfaces described elsewhere in this chapter 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 is identical to the Elliptical Grating Surface, see the "Elliptical Grating surface shape" on page 242 for details.



### 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 directory 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 of the elliptical grating surface type.
- 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.
- 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 optics must be of the elliptical grating 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 the Chapter "Optimization" for information about the CMFV operand.

### 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 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6
Holo Type	Diffraction order	a	b	c	OPD Mode

**Paraxial**

The paraxial surface acts as an ideal thin lens. The paraxial surface is particularly useful for analyzing and optimizing systems whose output is collimated, or nearly collimated light. Afocal systems can be modeled by placing a paraxial surface before the image plane, and then setting the thickness of the paraxial surface (the distance to the image plane) to be the same as the focal length. If the focal length is chosen to be one meter, then all aberration data presented in micrometers (such as ray fans) can be interpreted to be in units of microradians.

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. 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 can be more difficult, especially in the presence of significant aberrations, as discussed below.

If the OPD mode is set to zero (the default), then the OPD computation is based upon the unaberrated conjugate positions. This is preferred for optical systems with modest aberrations which are well described by first-order optics.

If the aberrations are large (more than 10 waves), or the system is not well described by first-order optics, then the paraxial lens OPD computation cannot assume that the lens is working at fixed conjugates for all incoming rays. For these systems, the mode should be set to 1. In this mode, the OPD is computed based upon the ray bending of the aberrated input beam. Using mode = 1 requires ZEMAX to iteratively integrate the actual phase introduced by the surface. This results in mode = 1 being substantially slower than mode = 0, and mode = 1 is not more accurate if the aberrations are reasonably small (less than 5-10 waves). When using mode = 1, for maximum OPD accuracy, the working F/# of the paraxial lens should be no faster than about F/4.

 ***Mode = 1 is substantially slower than mode = 0!***

The paraxial surface refracts rays using the following equations:

$$n'u_x' = nu_x - x\phi$$

$$n'u_y' = nu_y - y\phi$$

where  $\phi$  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” on page 87.

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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} - A \left[ \frac{1}{4} [1 + \cos(2\pi\alpha x)][1 + \cos(2\pi\beta y)] - 1 \right],$$

where  $A$  is the peak to valley height in lens units of the modulation, and the values  $\alpha$  and  $\beta$  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
$A$	$\alpha$	$\beta$

## **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" on page 247.

### PARAMETER DEFINITIONS FOR POLYNOMIAL SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
$\gamma_1$	$\gamma_2$	$\gamma_3$	$\gamma_4$	$\gamma_5$	$\gamma_6$	$\gamma_7$	$\gamma_8$

## **Radial Grating**

The radial grating surface is similar to the diffraction grating surface (see "Diffraction Grating" on page 241), 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" on page 243). 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),  $\theta_2$  is the refracted angle,  $\theta_1$  is the incident angle, M is the diffraction order,  $\lambda$  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:

$$d(p) = A_0 + A_1 p^1 + A_2 p^{-1} + A_3 p^2 + A_4 p^{-2} + \dots,$$

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

Extra Data Number	Description
1	Maximum number of terms
2	Normalization Radius
3	Coefficient on $p^0$
4	Coefficient on $p^1$
n	Coefficient on $p^{(n-3)}$

### **Radial NURBS**



*This feature is only available in the EE edition 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  $z = 0$ , 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.

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

Extra Data Number	Description
1	Number of control points. At least 4 points are required, and no more than 60 allowed.
2	y coordinate for control point 1.
3	z coordinate (sag) for control point 1.

Extra Data Number	Description
4	w (weight) value for control point 1.
3n-1, 3n, 3n+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.

This surface uses no parameter or extra data values.

## **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

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}}$$

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" on page 31. 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

$$\frac{1}{c} = R = \pm \frac{b^2}{a},$$

$$k = -\epsilon^2 = -\left[\frac{a^2 - b^2}{a^2}\right].$$

### **Modeling an axicon with the standard surface**

The standard surface can be used to make an almost perfect axicon. If  $(1+k)c^2r^2 \gg 1$ , 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  $\alpha$  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 ( $\alpha$ ) 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.

## **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” on page 243 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

$$A = \frac{k}{R},$$

$$B = 1 + U_1s^2 + U_2s^4 + \dots, \text{ and}$$

$$C = \frac{s^2}{R} + V_1s^4 + V_2s^6 + \dots,$$

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

Extra Data Number	Description
1	Maximum term number. The maximum is 250, but 4-10 is typical.
2	$U_1$
3	$V_1$
even n	$U^{\frac{n}{2}}$
odd n	$V^{\frac{n-1}{2}}$

### **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. It 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:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14}.$$

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  $r$ . 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 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 chapter.

## PARAMETER DEFINITIONS FOR TOROIDAL SURFACES

Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Radius of Rotation	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$

### **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" on page 248.

## 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" on page 272. 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" on page 264.



***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	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$

#### EXTRA DATA DEFINITIONS FOR TOROIDAL HOLOGRAM SURFACES

Extra Data Number	Description
1	Construction coordinate X1 for first construction beam.
2	Construction coordinate Y1 for first construction beam.
3	Construction coordinate Z1 for first construction beam.
4	Construction coordinate X2 for second construction beam.
5	Construction coordinate Y2 for second construction beam.
6	Construction coordinate Z2 for second construction beam.
7	Construction wavelength.
8	Playback order.
9	Converge/Diverge flag.

### **Toroidal NURBS**



***This feature is only available in the EE edition of ZEMAX.***

The toroidal NURBS surface is similar to the radial NURBS surface (page 269). 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

Extra Data Number	Description
1	Number of control points. At least 4 points are required, and no more than 60 allowed.
2	y coordinate for control point 1.
3	z coordinate (sag) for control point 1.
4	w (weight) value for control point 1.
3n-1, 3n, 3n+1	y, z, and w for control point n.

### **User Defined**



***This feature is only available in the EE edition 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 file is essentially 2 functions;

the first is "DLLMain" which is used to (optionally) initialize data used by the DLL. The other function is "UserDefinedSurface", and this is the function that is generally modified for each surface type. The function name cannot be changed; ZEMAX looks for a function of this name. The function may also be defined as "UserDefinedSurface2"; this function definition uses a newer, larger data structure that supports additional features not supported in the original definition of "UserDefined Surface".

ZEMAX passes two structures to the function, and these structures are defined in the USERSURF.H file. The USERSURF.H file defines two separate "fixed data" structures, one used by UserDefinedSurface and the other by UserDefinedSurface2. Inside the UserDefinedSurface or UserDefinedSurface2 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	<p>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.</p> <p>The DLL only needs to handle this type code if the DLL requires one-time initialization before the DLL can trace rays.</p>
9	<p>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.</p> <p>The DLL only needs to handle this type code if the DLL must release memory or perform some other "clean-up" before terminating.</p>
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 directory \ZEMAX\DLL, 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.

### Diffraction UDS DLLs

Diffraction 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 diffraction optics is defined by:

$$l' = l + \frac{\lambda}{2\pi} \frac{\partial \phi}{\partial x}$$

$$m' = m + \frac{\lambda}{2\pi} \frac{\partial \phi}{\partial y}$$

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  $\phi$  is the phase in radians.

A sample DLL, US\_GRATE.DLL, illustrates the diffraction computation by cloning the ZEMAX grating surface.

### Lenslet arrays using UDS 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 (TIR's) 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
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 $T(x, y) = \exp\left(-2G_x\left(\frac{x}{R}\right)^2 - 2G_y\left(\frac{y}{R}\right)^2\right),$ <p>where R is the semi-diameter of the surface and x and y are the coordinates at which the ray intercepts the surface.</p>
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.
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.
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.

DLL name	Description
US_FILT1	This surface defines an apodization function of the form $T = 1.0 - \text{Exp}(-G\rho^2)$ where $\rho$ 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 $D_{\text{max}}$ from zero to a normalized radial coordinate $\rho_1$ , then linearly goes to $D=0$ at $\rho_2$ , and $D=0$ thereafter. The density is $D = -\text{Log}_{10}(T)$ where $T$ is the transmission.
US_FILT3	This surface defines an apodization function of the form $D = 0.5D_{\text{max}}(1.0 + \cos\pi\rho)$ where $\rho$ is the radial coordinate normalized to the semi-diameter of the surface.
US_FILT4	<p>This surface has an apodization function defined by:</p> $T(r) = 1 \text{ if } r < R - \Delta R,$ $T(r) = 0 \text{ if } r > R + \Delta R,$ $T(r) = \frac{1 + \cos\left(\frac{\pi(r - (R - \Delta R))}{2\Delta R}\right)}{2} \text{ if } R - \Delta R < r < R + \Delta R,$ <p>where <math>r</math> is the radial coordinate, <math>R</math> is the aperture radius, and <math>\Delta</math> is a dimensionless fraction of the aperture radius. The transmission varies smoothly from 1 to 0 over twice the distance defined by <math>\Delta R</math>. The value for <math>\Delta</math> is defined as 0.001 if the provided value for <math>\Delta</math> is zero. This function makes a convenient "soft-edged" aperture which is useful for optimization of transmission through a circular aperture.</p>
US_FILT5	<p>This surface has an apodization function defined by:</p> $T(na) = 1.0 \text{ if } na < NA - \Delta,$ $T(na) = 0.0 \text{ if } na > NA + \Delta, \text{ otherwise}$ $T(na) = 0.5\left(1 + \cos\left(\frac{(na - (NA - \Delta NA))\pi}{(2\Delta NA)}\right)\right),$ <p>where <math>na</math> is the numerical aperture of the incoming ray, <math>NA</math> is a user defined parameter, and <math>\Delta</math> 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 its incident angle, rather than position. Rays whose <math>na</math> exceeds the <math>NA - \Delta</math> value will be attenuated smoothly to zero. This permits easier, more continuous optimization.</p>
US_FILT6	<p>This surface has an apodization function defined by:</p> $T(x, y) = (1/4)(1 + \sin(2\pi\alpha x))(1 + \sin(2\pi\beta y)),$ <p>where <math>\alpha</math> and <math>\beta</math> are the spatial frequencies of the sinusoidal transmission in <math>x</math> and <math>y</math>, respectively.</p>

DLL name	Description
US_FILT7	This surface apodizes the surface transmission via linear interpolation of values specified on the extra data editor along the radial coordinate.
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.
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	<p>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:</p> <p>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)...</p> <p>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.</p> <p>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:</p> <pre> 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.. </pre> <p>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.</p>
US_OFFST	<p>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.</p>
US_OGIVE	<p>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</p> $z = \frac{cr_g^2}{1 + \sqrt{1 - (1+k)c^2r_g^2}}, \text{ where}$ $r_g = r_o + \sqrt{x^2 + y^2}.$

DLL name	Description
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  $\lambda_0$ , the construction wavelength. This is the wavelength at which the grating will reflect the axis ray along the angle  $\beta_0$ .

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
$M$	$L$	$\cos \alpha_0$	$\cos \beta_0$	$\lambda_0$

## **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  $i^{th}$  Zernike Fringe polynomial,  $\rho$  is the normalized radial ray coordinate,  $\phi$  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” on page 165. ZEMAX supports up to 37 terms. The coefficients  $A_i$  all have units of waves. One wave is  $2\pi$  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” on page 283. See also “Zernike Standard Phase” on page 284.

## PARAMETER DEFINITIONS FOR ZERNIKE FRINGE PHASE SURFACES

Parameter 0	Parameter 1
Diffraction order	Extrapolate

## EXTRA DATA DEFINITIONS FOR ZERNIKE FRINGE PHASE SURFACES

Extra Data Number	Description
1	Number of terms.
2	Normalization radius. Coordinates are normalized by this value.
3 - 39	Coefficients on Zernike polynomials 1 - 37, respectively, in units of primary waves.
40 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” on page 230 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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^8 \alpha_i r^{2i} + \sum_{i=1}^N A_i Z_i(\rho, \phi)$$

where  $N$  is the number of Zernike coefficients in the series,  $A_i$  is the coefficient on the  $i^{th}$  Zernike Fringe polynomial,  $r$  is the radial ray coordinate in lens units,  $\rho$  is the normalized radial ray coordinate, and  $\phi$  is the

angular ray coordinate. The Zernike Fringe polynomials are defined in the table “Zernike Fringe Coefficients” on page 165. 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  $\alpha_i$  have units, and are as defined in the section “Even Asphere” on page 243. 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” on page 282. Also see “Zernike Standard Sag” on page 285.

#### PARAMETER DEFINITIONS FOR ZERNIKE FRINGE SAG SURFACES

Parameter 0	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Extrapolate	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$

#### EXTRA DATA DEFINITIONS FOR ZERNIKE FRINGE SAG SURFACES

Extra Data Number	Description
1	Number of terms.
2	Normalization radius. Coordinates are normalized by this value.
3 - 39	Coefficients on Zernike polynomials 1 - 37, respectively, in lens units.
40 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

$$\Phi = M \sum_{i=1}^N 2\pi A_i Z_i(\rho, \phi)$$

where  $N$  is the number of Zernike coefficients in the series,  $A_i$  is the coefficient on the  $i^{th}$  Zernike Standard polynomial,  $\rho$  is the normalized radial ray coordinate,  $\phi$  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” on page 168. ZEMAX supports up to 231 terms. The coefficients  $A_i$  all have units of waves. One wave is  $2\pi$  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" on page 285.

### PARAMETER DEFINITIONS FOR ZERNIKE STANDARD PHASE SURFACES

Parameter 0	Parameter 1
Diffraction order	Extrapolate

### EXTRA DATA DEFINITIONS FOR ZERNIKE STANDARD PHASE SURFACES

Extra Data Number	Description
1	Number of terms.
2	Normalization radius. Coordinates are normalized by this value.
3 - 233	Coefficients on Zernike polynomials 1 - 231, respectively, in units of primary waves.
234 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" on page 230 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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^8 \alpha_i r^{2i} + \sum_{i=1}^N A_i Z_i(\rho, \varphi)$$

where  $N$  is the number of Zernike coefficients in the series,  $A_i$  is the coefficient on the  $i^{th}$  Zernike Standard polynomial,  $r$  is the radial ray coordinate in lens units,  $\rho$  is the normalized radial ray coordinate, and  $\varphi$  is the angular ray coordinate. The Zernike Standard polynomials are defined in the table given in "Zernike Standard Coefficients" on page 168. 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  $\alpha_i$  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	Parameter 1	Parameter 2	Parameter 3	Parameter 4	Parameter 5	Parameter 6	Parameter 7	Parameter 8
Extrapolate	$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$	$\alpha_6$	$\alpha_7$	$\alpha_8$

### EXTRA DATA DEFINITIONS FOR ZERNIKE STANDARD SAG SURFACES

Extra Data Number	Description
1	Number of terms.
2	Normalization radius. Coordinates are normalized by this value.
3 - 233	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.

## **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 250 zones may be defined. Three parameter values are required: the mode, the  $\Delta r$ , and the reference wavelength,  $\lambda_0$ .

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 " $\Delta r$ " 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 media.

The optical thickness is defined at  $n$  points, separated by  $\Delta r$ :  $r_1, r_2, r_3, \dots, r_n$ . The number of points, " $n$ ", is specified in the first column of the Extra Data Editor. The remaining terms are defined in columns 2 through  $n+1$  in the Extra Data Editor.

#### PARAMETER DEFINITIONS FOR ZONE PLATE SURFACES

Parameter 1	Parameter 2	Parameter 3
Mode	$\Delta r$	Ref. Wave

#### EXTRA DATA DEFINITIONS FOR ZONE PLATE SURFACES

Extra Data Number	Description
1	Number of points
2	Optical thickness at $\Delta r$
3	Optical thickness at $2 \Delta r$
4	Optical thickness at $3 \Delta r$
$n+1$	Optical thickness at $n \Delta r$





## **Introduction**



*This feature is only available in the EE edition of ZEMAX.*

---

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. 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 faceted objects, 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.

This chapter provides information on setting up a NSC group, defining objects, and tracing rays through NSC.

## **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 faceted parts, obscurations, and 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.

## **Optimization with NSC**

See “Optimizing objects in a non-sequential group with sequential rays” on page 427 and “Optimizing with sources and detectors in non-sequential mode” on page 427.

### **The two 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**

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 "Overview of NSC ray tracing with ports" below. 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**

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.

#### **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.

## **Overview of NSC ray tracing 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  $4\pi$  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.

### **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 (see page 239).

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.

The diameter of the circular exit port in lens units. *This value 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; although this is the default value when the non-sequential surface is first created.

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 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.

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 ("Glue Distance In Lens Units" on page 95) of the entrance port; otherwise rays entering the NSC group will immediately strike the exit port and not intersect any of the objects within.

### 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.

### Tracing rays within the NSC

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" on page 94).

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.

## **Overview of NSC ray tracing without ports**

Ray tracing through a group of NSC objects without using ports is accomplished with the following basic steps:

- 1) Choose File, Non-Sequential Mode from the main menu.
- 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 (on the wavelength data editor)

The glass catalogs to be used (on the general system data dialog box)

The coating definitions (in the coating file associated with the current lens)

### 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. See “Choosing the number of analysis rays” on page 365.

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” on page 87 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.

### Tracing rays within the NSC

Once inside the NSC group, 2 things can happen to a ray:

- 1) It can hit nothing at all.
- 2) It can hit one of the objects within the group.

If the ray hits nothing at all, then the ray tracing for this ray is terminated.

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 on the System dialog box, see “Non-Sequential” on page 94. 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.

### Launching rays for analysis

Rays may be launched, and detectors reset, from the Detectors, Ray Trace/Detector Control.

## **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. For details on reflective, refractive, and absorptive properties, see a following section.

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 “Object Placement” on page 344 for information on placing objects inside or adjacent to one another.



***Detectors are described on page 332. Sources are described on page 336.***

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 OF NSC OBJECTS

Object Name	Description	Page
Annular Aspheric Lens	An annular volume with an aspheric surface on one face.	297
Annular Volume	A volume formed by two tilted, elliptical annular planar faces.	297
Annulus	A planar ellipse with an elliptical hole in the middle.	298
Aspheric Surface	A surface with a conic substrate plus even and odd radial polynomial power aspheric terms.	298
Aspheric Surface 2	Identical to the Aspheric Surface, with support for more complex aperture types.	299
Axicon Surface	An axicon formed by rotating a spherical arc around the z axis.	299
Biconic Lens	A circular or rectangular lens with separate curvature and conic in the X and Y directions on both front and back faces.	300
Biconic Surface	A surface with separate curvature and conic in the X and Y directions. May be partially hyperhemispheric under special circumstances.	301
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.	302
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.	303
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.	303
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.	304
Compound Parabolic Concentrator (CPC)	A compound parabolic concentrator (CPC). The CPC may be hollow or solid.	305
Cylinder Pipe	A tapered cylindrical surface shape.	305
Cylinder Volume	A tapered cylindrical volume, with end caps.	306
Cylinder 2 Pipe	A cylindrical surface shape with tilted ends.	306
Cylinder 2 Volume	A cylindrical volume, with tilted end caps.	306
Diffraction Grating	A standard lens with a diffraction grating with constant line spacing on the front face.	307
Ellipse	A planar ellipse.	307

Object Name	Description	Page
Elliptical Volume	An elliptical volume (or shell). The shape is a tapered cylinder with an elliptical cross section.	307
Even Asphere Lens	A rotationally symmetric lens with up to 16th order aspheres on both the front and back faces.	308
Extended Polynomial Lens	A lens with extended polynomial surfaces on front and back faces.	308
Extended Polynomial Surface	A surface with an extended polynomial shape.	309
Fresnel 1	A true exact (not faceted) Fresnel lens, either radial or cylindrical, defined in terms of a number of construction parameters.	309
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.	310
Hexagonal Lenslet	A lenslet array with hexagonal symmetry.	312
Hologram Lens	A circular or rectangular solid with an optically fabricated hologram on the front face.	312
Hologram Surface	A circular or user defined plane, conic, spherical or aspherical surface with an optically fabricated hologram on the face.	313
Imported	An object or objects imported from a CAD program in either IGES or STEP format.	314
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.	315
Lenslet 1	A rectangular shaped lens object with a curved back face. This object can be used to model an array element, and may be diffractive.	315
Lenslet 2	A rectangular shaped lens object with curved front and back faces. This object can be used to model an array element.	317
MicroElectroMechanical System (MEMS)	An array of flat facets (intended to be made of material "MIRROR") where the individual facets may be tipped about any of 3 angles. The facets may be addressed either by rows, by columns, or by individual pixels.	317
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.	318
Odd Asphere Lens	A rotationally symmetric lens with up to 12th order odd and even power aspheres on both the front and back faces.	318
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.	319
Rectangular Corner	A corner composed of 3 rectangular surfaces.	319
Rectangle	A rectangle defined by a width and height.	320
Rectangular Pipe	A four sided box.	320
Rectangular Pipe Grating	A four sided box with a grating on the side faces.	320
Rectangular Roof	A roof composed of two rectangular surfaces meeting at an angle.	321

Object Name	Description	Page
Rectangular Torus Surface	A section of a rectangular torus which defines a surface; open at the ends.	321
Rectangular Torus Volume	A section of a rectangular torus with closed ends to define a volume.	321
Rectangular Volume	A six sided box enclosing a volume.	322
Rectangular Volume Grating	A six sided box with a diffraction grating on side faces.	322
Slide	A color RGB file, in either BMP or JPG 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.	322
Sphere	A sphere.	323
Standard Lens	A 5 surface lens object with two Standard faces, two flat edges, and a cylindrical outer edge.	323
Standard Surface	A surface with the same shape as the ZEMAX Standard surface (a plane, sphere, conic asphere, or hyperhemisphere).	324
STL Object	A general polygon object defined in STL format, typically exported from a CAD program.	324
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.	325
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.	326
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.	327
Toroidal Lens	A circular, elliptical, or rectangular lens with aspheric toroidal surfaces on the front and back sides.	327
Toroidal Surface	A rectangular surface with an aspheric toroidal shape.	327
Torus Surface	A section of a torus which defines a surface; open at the ends.	328
Torus Volume	A section of a volume torus; like a surface but with end caps to enclose a volume.	328
Triangular Corner	A corner composed of 3 triangles.	329
Triangle	A triangle defined by 3 points in a plane.	329
User Defined Object	An object defined by a user provided DLL.	329
Zernike Surface	A surface defined by radial aspheric and standard Zernike polynomial sag terms.	331

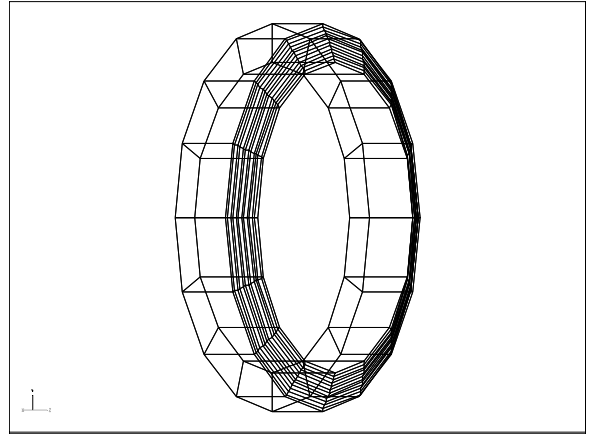
Most objects require specification of various parameters. The parameters required by each object type and detailed descriptions of each object follow.



## Annular Aspheric Lens

The annular aspheric lens is an annular solid with a plano front face. The rear face is defined by a conic asphere with even aspheric polynomial terms to 16th order, identical to the Even Asphere Lens object (see "Even Asphere Lens" on page 308). The annular shape is defined by a min and max radial aperture. The aspheric rear face is offset from the front by a thickness measured at a specified radial aperture. The object is defined by the following parameters:

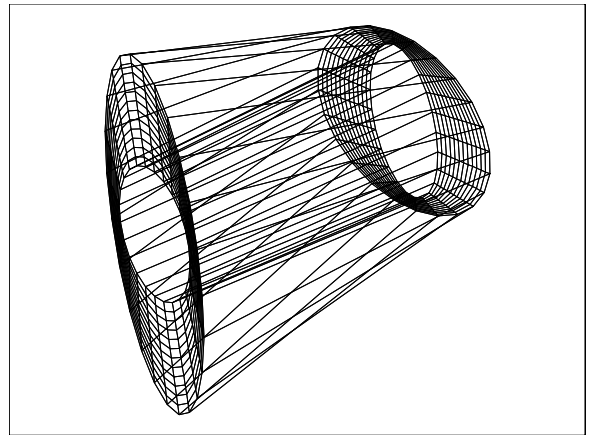
1. The number of angular facets
2. The number of radial facets
3. The Minimum Radial Aperture
4. The Maximum Radial Aperture
5. The Thickness Aperture, which is the radial aperture to measure the object thickness
6. Radius
7. Conic
8. Thickness (measured at the Thickness Aperture)
- 9-16. The Even Aspheric coefficients  $\alpha_2$  through  $\alpha_{16}$ .



## Annular Volume

The annular volume consists of tilted annular elliptical faces front and rear, separated by an axial distance. The defining parameters are:

- 1: The number of angular facets. See "The use of facets" on page 331.
- 2: The number of radial facets. See "The use of facets" on page 331.
- 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.
- 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.



Coating/Scatter Groups: Outside face connecting the "maximum" apertures CSG #0, front face CSG #1, back face CSG #2, inside face connecting the "minimum" apertures on the front and back CSG #3.

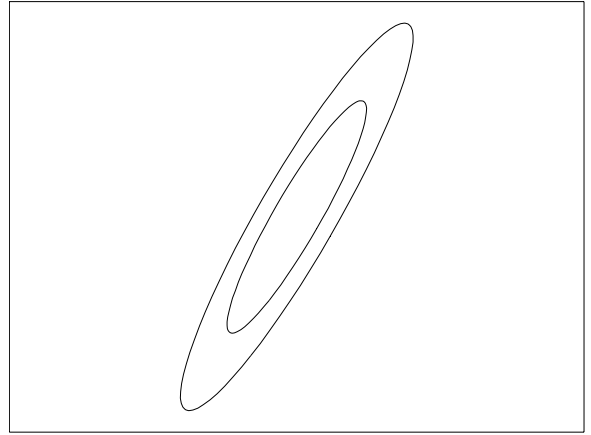
## Annulus

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. Coating/Scatter Groups: All faces CSG #0.



## Aspheric Surface

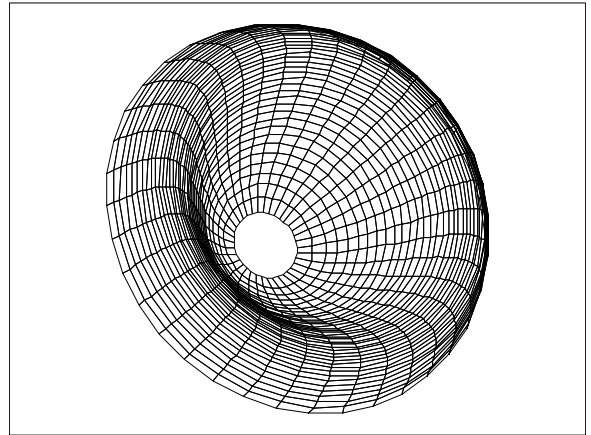
An aspheric surface is defined by the following sag equation:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^N \alpha_i r^i,$$

where  $c$  is the curvature of the surface,  $k$  is the conic constant,  $r$  is the radial coordinate, and the  $\alpha$  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: The number of angular facets. See "The use of facets" on page 331.
- 6: The number of radial facets. See "The use of facets" on page 331.
- 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.
- 8-250: The  $\alpha$  coefficients on the polynomial expansion.

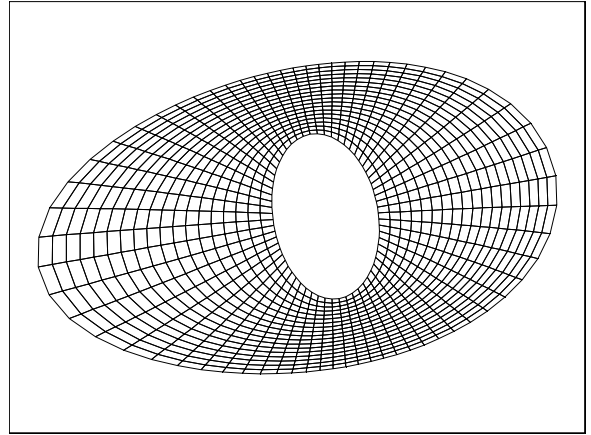


Coating/Scatter Groups: All faces CSG #0. This object supports user defined apertures, see "User defined apertures" on page 352. For other aperture shape options, see "Aspheric Surface 2" on page 299.

## 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: The number of angular or X facets. See "The use of facets" on page 331.
  - 11: The number of radial or Y facets. See "The use of facets" on page 331.
  - 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.
  - 13-250: The  $\alpha$  coefficients on the polynomial expansion.
- Coating/Scatter Groups: All faces CSG #0.



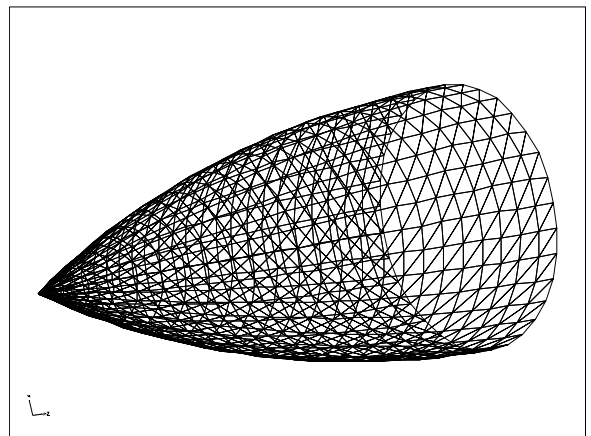
## Axicon Surface

There are many types of axicons. Common polynomial types can be modeled using the Aspheric Surface (see "Aspheric Surface" on page 298).

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.

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.
  - 4: The number of angular facets. See "The use of facets" on page 331.
  - 5: The number of z direction facets. See "The use of facets" on page 331.
- Coating/Scatter Groups: All faces CSG #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: The maximum number of facets in the X direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of angular facets. See “The use of facets” on page 331.

5: The maximum number of facets in the Y direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of radial facets. See “The use of facets” on page 331.

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.

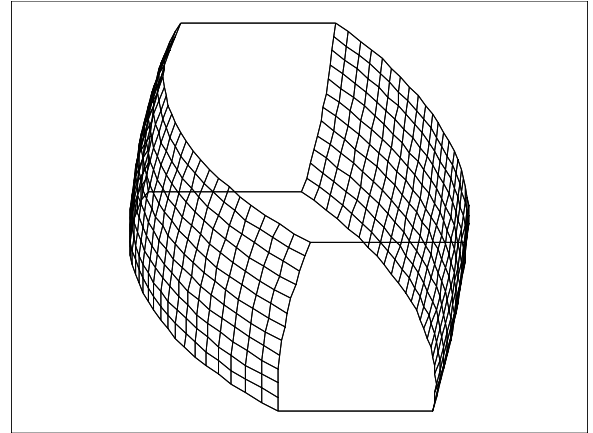
10: The base radius of curvature in the XZ plane for the back surface. If this value is zero, then the XZ curvature is assumed to be zero.

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.

Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #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: The maximum X aperture in lens units.

6: The maximum Y aperture in lens units.

7: The minimum X aperture in lens units. This parameter is ignored if the surface has elliptical symmetry.

8: The minimum Y aperture in lens units. This parameter is ignored if the surface has elliptical symmetry.

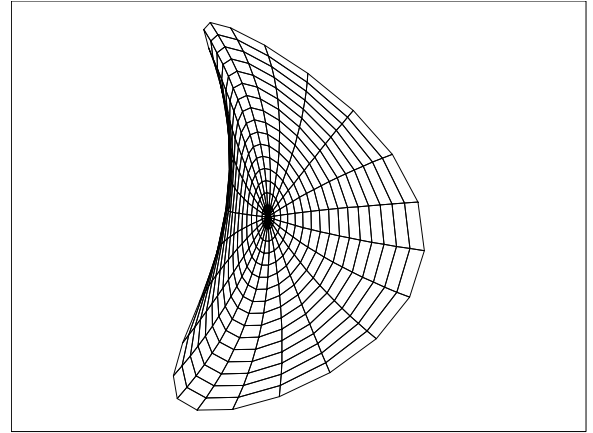
9: The maximum number of facets in the X direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of radial facets. See "The use of facets" on page 331.

10: The maximum number of facets in the Y direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of angular facets. See "The use of facets" on page 331.

11: The "Is Rectangle?" flag. If zero, the resulting surface shape will be elliptical. 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 surface is elliptical, 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.

Coating/Scatter Groups: All faces CSG #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  $i^{th}$  extended polynomial term. The polynomials are a power series in the normalized coordinates  $x$  and  $y$ , as described in "Extended Polynomial Lens" on page 308. The coefficients  $A_i$  all have units of radians ( $2\pi$  radians is one wave).

The definitions for the parameters are:

1-9: See the Standard Lens description for information on these parameters.

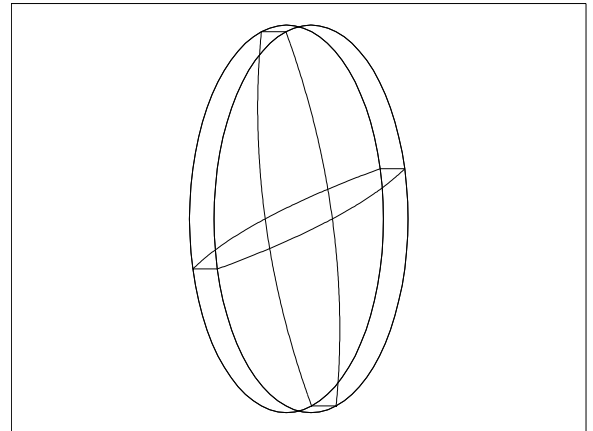
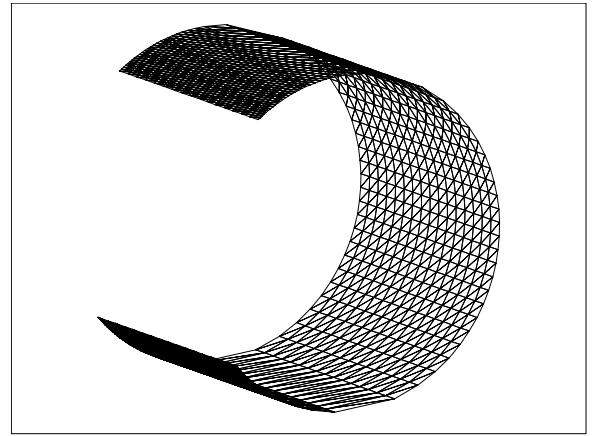
10: The diffraction order  $M$ .

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” on page 347.

Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #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  $\rho$ , which is the normalized radial aperture coordinate. The coefficients  $A_i$  all have units of radians ( $2\pi$  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.

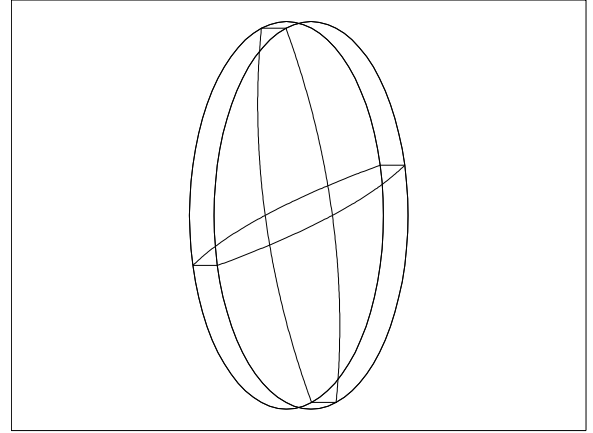
12: The maximum term number.

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” on page 347.

Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \sum_{i=1}^{12} \alpha_i r^{2i}.$$

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  $\rho$ , which is the normalized radial aperture coordinate. The coefficients  $A_i$  all have units of radians ( $2\pi$  radians is one wave).

The definitions for the parameters are:

- 1: The Maximum Radial aperture.
- 2: The Thickness of the lens at the center.
- 3: The number of angular Facets. See “The use of facets” on page 331.
- 4: The number of radial Facets. See “The use of facets” on page 331.
- 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” on page 347. The reference coordinate is the center of the front face. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

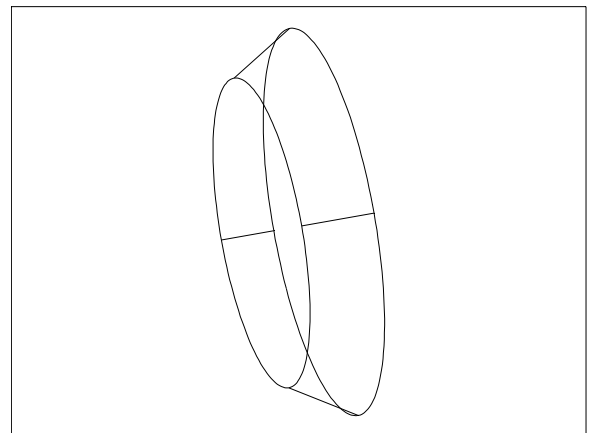
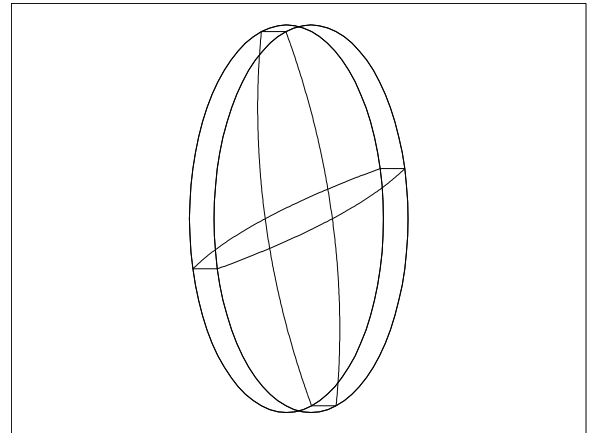
## Cone

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 reference point. Coating/Scatter Groups: All faces CSG #0.





## Compound Parabolic Concentrator (CPC)

A CPC is defined by 6 parameters:

- 1: The radial aperture at  $z = 0$ .
- 2: The maximum acceptance angle in degrees.
- 3: The length along the local Z axis.
- 4: The number of angular facets. See "The use of facets" on page 331.
- 5: The number of length facets. See "The use of facets" on page 331.
- 6: The "Is Volume?" flag.

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. The maximum value for the CPC length is given by

$$L = \frac{a(1 + \sin\theta)}{\tan\theta \sin\theta},$$

where  $a$  is the radial aperture and  $\theta$  is the acceptance angle. Longer lengths will be truncated to this value. Coating/Scatter Groups: Front face CSG #1, rear face CSG #2, outer faces CSG #0.

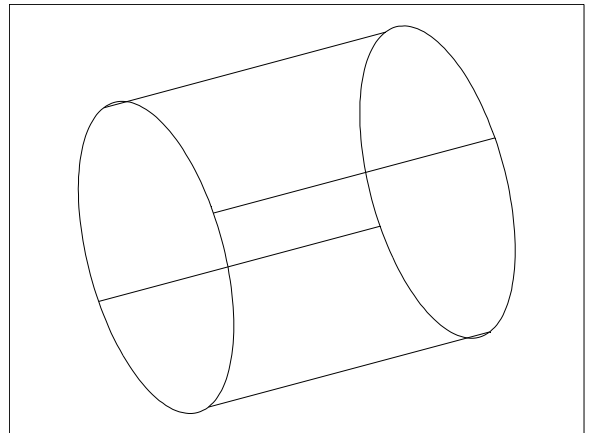
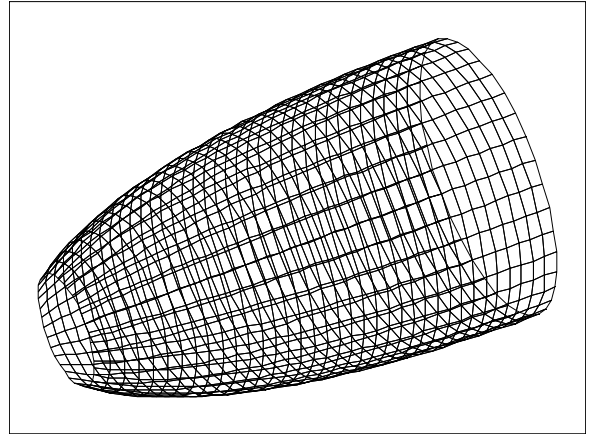
## Cylinder Pipe

A cylinder pipe is a rotationally symmetric surface defined by 3 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. Coating/Scatter Groups: All faces CSG #0.



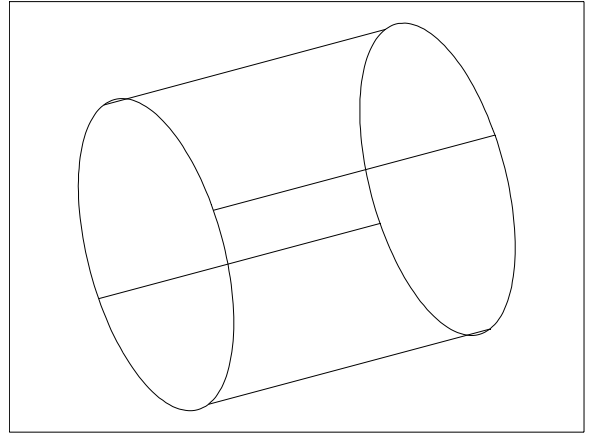
## Cylinder Volume

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. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

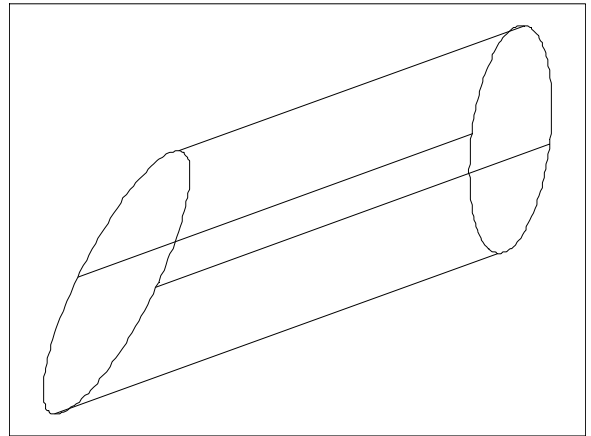


## Cylinder 2 Pipe

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. Coating/Scatter Groups: All faces CSG #0.



## Cylinder 2 Volume

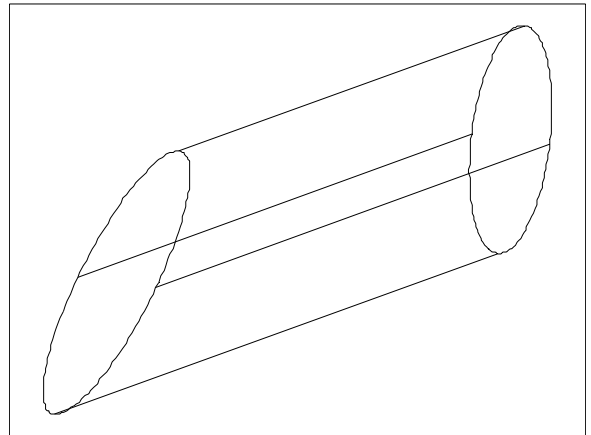
A cylinder volume is a rotationally symmetric volume defined by 4 parameters:

- 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. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.



## Diffraction Grating

The diffraction grating is very similar to a Standard Lens, with 2 additional parameters:

1-9: See the Standard Lens description for information on these parameters.

10: The grating line frequency in lines/micrometer on the front face.

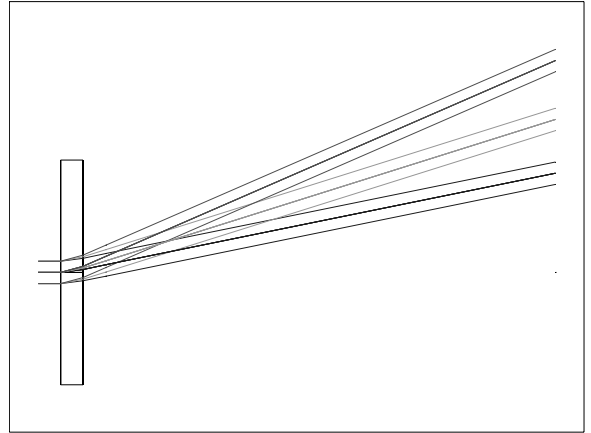
11: The diffraction order for the front face.

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.

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" on page 347.

Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.



## Ellipse

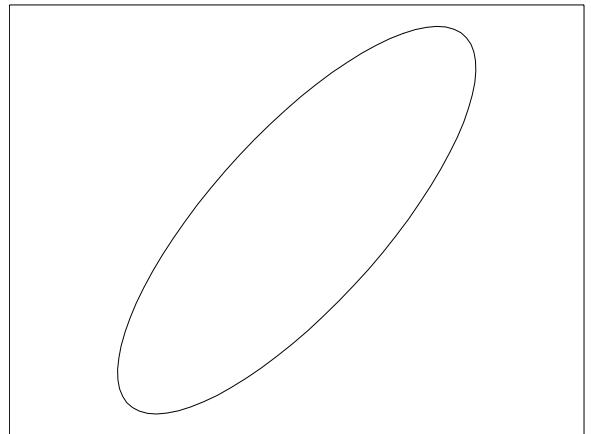
The ellipse is a plane elliptical surface shape defined by 2 parameters:

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. Coating/Scatter Groups: All faces CSG #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.

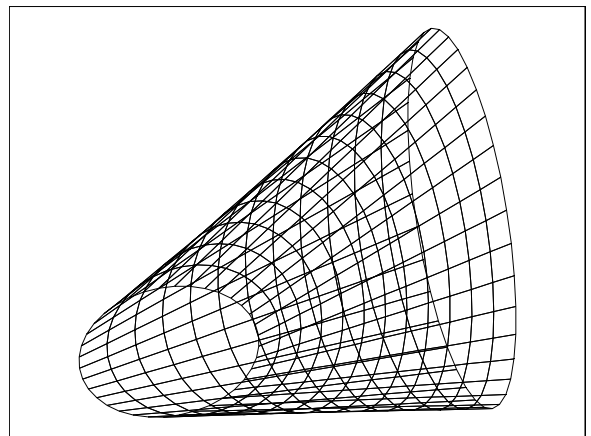
6: The number of z facets. See "The use of facets" on page 331.

7: The number of angular facets. See "The use of facets" on page 331.

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. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.



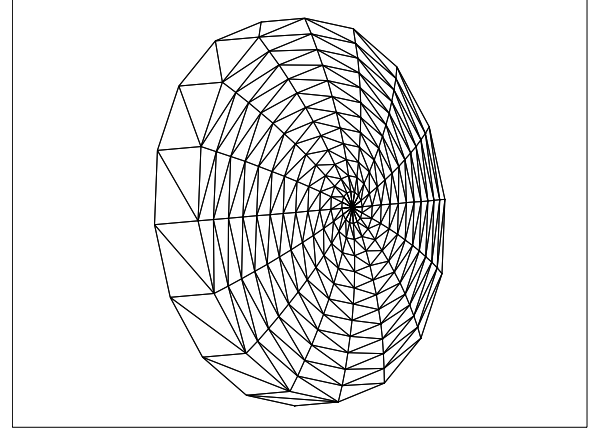
## Even Asphere Lens

The Even Asphere surface shape is defined by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16},$$

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: The number of angular Facets. See “The use of facets” on page 331.
- 4: The number of radial Facets. See “The use of facets” on page 331.
- 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. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

## Extended Polynomial Lens

The Extended Polynomial surface shape is defined by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \sum_{i=1}^N A_i E_i(x, y).$$

where  $N$  is the number of polynomial coefficients in the series, and  $A_i$  is the coefficient on the  $i^{th}$  extended polynomial term. The polynomials are a power series in  $x$  and  $y$ . The first term is  $x$ , then  $y$ , then  $x^2$ ,  $x^2y$ ,  $y^2y$ , 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: The maximum number of facets in the X direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of angular facets. See “The use of facets” on page 331.

5: The maximum number of facets in the Y direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of radial facets. See “The use of facets” on page 331.

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. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #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 if circular. This object supports user defined aperture shapes, see “User defined apertures” on page 352.

2: The radius of curvature.

3: The conic constant k.

4: The number of angular Facets. See “The use of facets” on page 331.

5: The number of radial Facets. See “The use of facets” on page 331.

6: The normalization radius.

7: The number of extended polynomial terms.

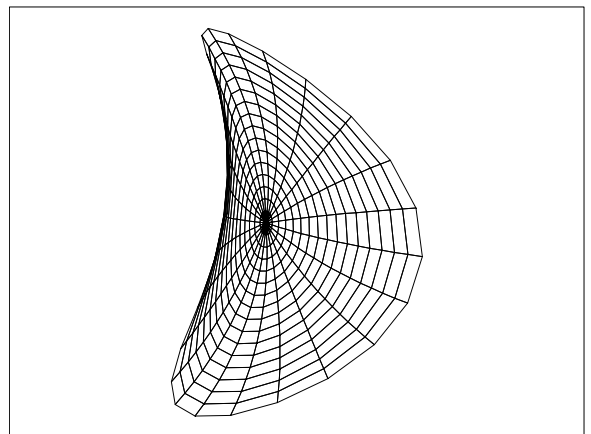
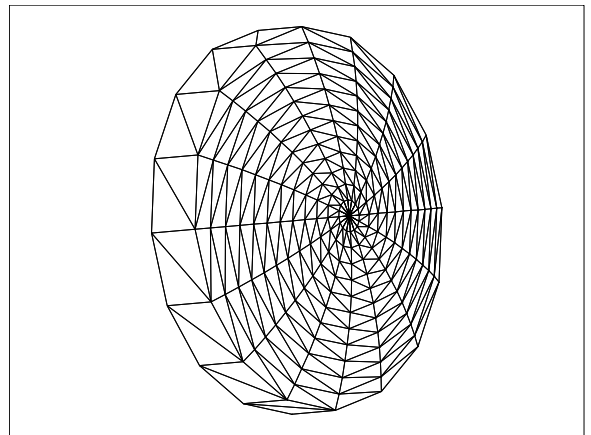
8 and on: The extended polynomial coefficients.

The reference coordinate is the center of the face. Coating/Scatter Groups: All faces CSG #0.

### Fresnel 1

This object is a general radially symmetric or cylindrical solid Fresnel lens constructed by modeling the actual facets and 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 facets defining the profile of the Fresnel that yields optical power. The profile is constructed



of radially flat facets (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 facet is offset just enough along the z axis so all facets 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 facets 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 may be 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.

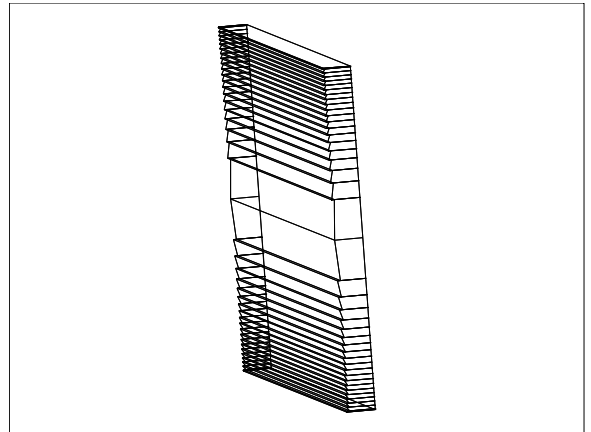
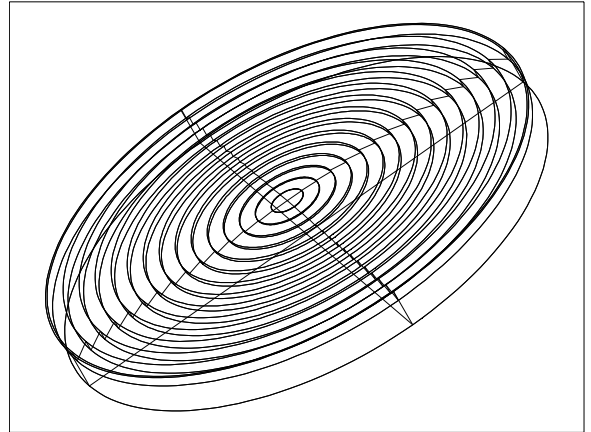
16: The number of sub segments. The greater the number of sub segments, the smoother the approximation to a curved surface between the grooves. A value of 1 yields flat grooves, larger values yield progressively more smooth curved faces at the expense of ray tracing speed.

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 facets. 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. Coating/Scatter Groups: All faces CSG #0.

## Fresnel 2

This object is an idealized Fresnel lens. Unlike the Fresnel 1, this Fresnel lens uses the approximation that the Fresnel facets are infinitesimally small, and may be ignored for purposes of computing the ray-object intersection point.



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:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1+k)c^2r^2}} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \alpha_4 r^8 + \alpha_5 r^{10} + \alpha_6 r^{12} + \alpha_7 r^{14} + \alpha_8 r^{16}.$$

If the Fresnel surface is cylindrical, then the profile is described by an identical expression in the y coordinate alone:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2y^2}} + \alpha_1 y^2 + \alpha_2 y^4 + \alpha_3 y^6 + \alpha_4 y^8 + \alpha_5 y^{10} + \alpha_6 y^{12} + \alpha_7 y^{14} + \alpha_8 y^{16}.$$

Note that 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.

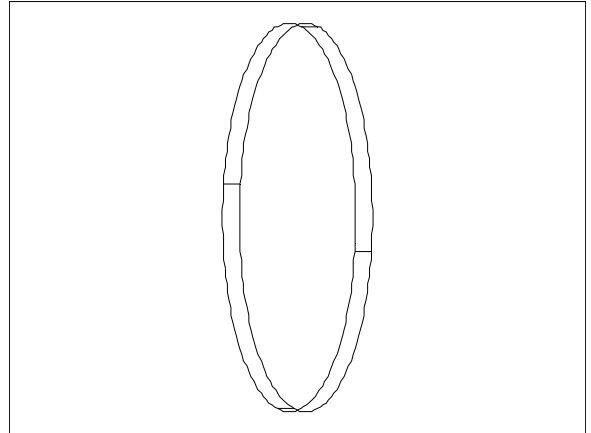
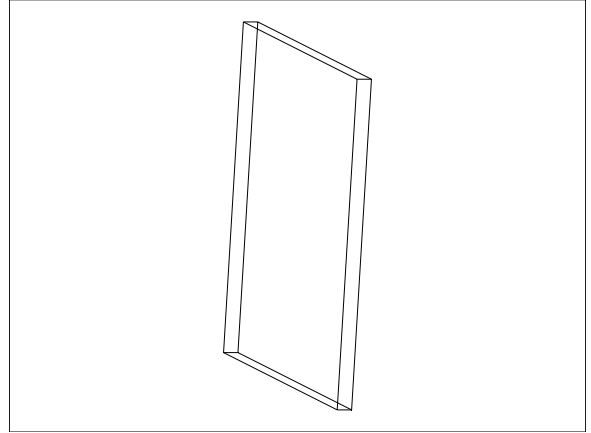
5: Radius: The base radius of curvature. This is one over the value "c" in the sag expressions above.

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 facets. Coating/Scatter Groups: All faces CSG #0.



## Hexagonal Lenslet

The hexagonal lenslet 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” on page 308). 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: # of facets. If this value is set to 0, 1, 2, 3, or 4, then 6, 24, 96, 384, or 1536 triangles respectively will be used to render each lenslet. Higher or lower values are internally set to either 4 or 0. Higher faceting yields a better looking picture, but has no affect on ray trace accuracy and may consume large amounts of memory and slow down both ray tracing and rendering. See “The use of facets” on page 331.
- 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.

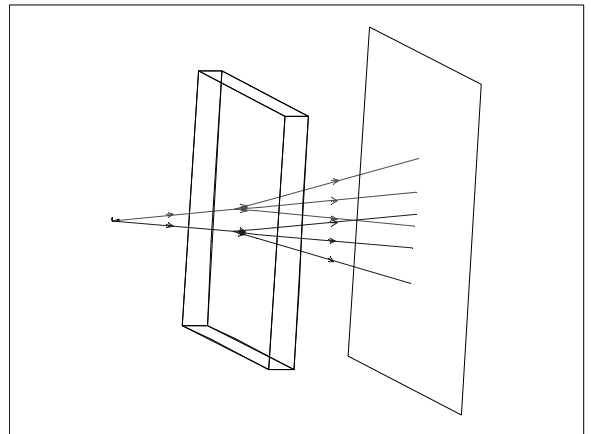
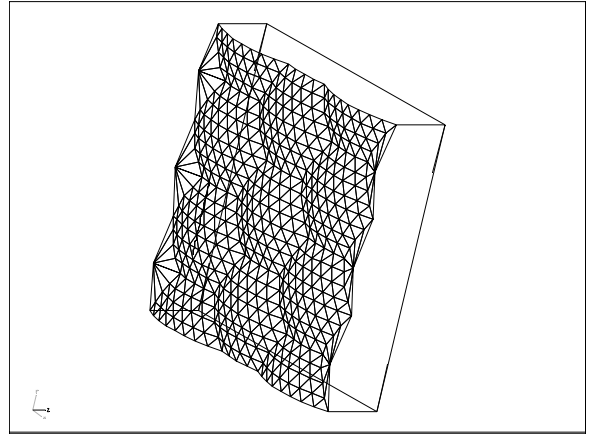
The reference coordinate is the center of the front face. Coating/Scatter Groups: Side faces CSG #0, front face CSG #1, back face CSG #2.

## Hologram Lens

This object is an ideal optically fabricated hologram similar to the Hologram 1 and Hologram 2 sequential surface models (see “Hologram 1” on page 261 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” on page 313.

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: The maximum number of facets in the X direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of angular facets. See “The use of facets” on page 331.
- 5: The maximum number of facets in the Y direction if the surface boundary is rectangular in shape, otherwise this is the maximum number of radial facets. See “The use of facets” on page 331.
- 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” on page 261.

11: Order: The diffraction order to use. Multiple orders may be specified, see “Diffraction tab” on page 350.

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” on page 347.

The reference coordinate is the center of the front face. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #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” on page 261 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” on page 312.

The surface shape is defined by the following sag equation:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^N \alpha_i r^i,$$

where  $c$  is the curvature of the surface,  $k$  is the conic constant,  $r$  is the radial coordinate, and the  $\alpha$  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” on page 298.

The following parameters are used to define the hologram 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: The number of angular facets. See “The use of facets” on page 331.

6: The number of radial facets. See “The use of facets” on page 331.

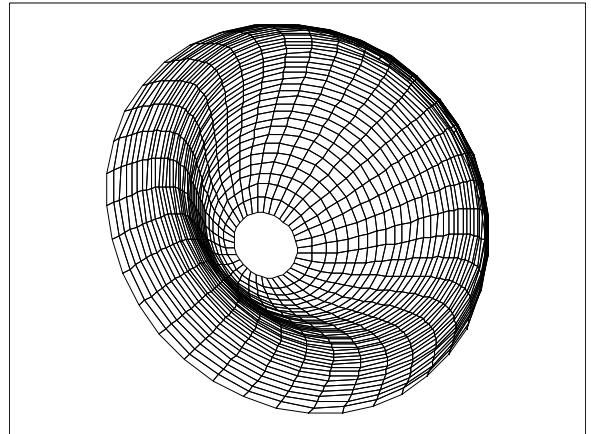
7: Holo Type: Use 1 for Hologram type 1 (both sources converging/diverging) or 2 for Hologram type 2 (one source converging, one diverging) See “Hologram 1” on page 261.

8: Order: The diffraction order to use. Multiple orders may be specified, see “Diffraction tab” on page 350.

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  $\alpha$  coefficients on the polynomial expansion.

Coating/Scatter Groups: All faces CSG #0. This object supports user defined apertures, see "User defined apertures" on page 352.

### Imported

Objects may be imported in either IGES, SAT, or STEP format. These file formats are commonly supported by CAD programs. 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 directory and end in the extension IGS, IGES, SAT, STP, or STEP. The extension must be included with the file name if entered 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 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.

3, 4, 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 10-10-10 if no other values seem obviously superior.

6) Chord Tolerance: This setting only affects the rendering of the solid. To render the solid, ZEMAX converts the solid to 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 solid. 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.

### 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; however, all optical properties will be identical for all the imported objects within a single file. To assign different optical properties (such as the material type) to each imported object, the objects must reside in separate files. 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" on page 329.

### 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 approximate the surface. This is probably adequate for mechanical design, but not for optical precision ray tracing, where surfaces must be known 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.

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 representation 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.

### Jones Matrix

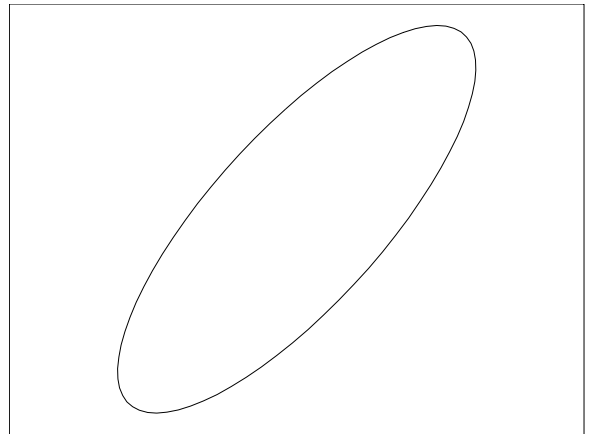
The Jones Matrix is a plane elliptical surface whose shape is defined by 2 parameters, and polarization transmission/reflection properties by 8 "ABCD" parameters:

- 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. This object is a special case of the more general annulus.

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" on page 511.

The reference coordinate is the center of the ellipse. Coating/ Scatter Groups: All faces CSG #0. This object supports user defined apertures, see "User defined apertures" on page 352.



### Lenslet 1

A lenslet 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.

The back surface may also be 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 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: The number of X and Y facets *for each lenslet*. See “The use of facets” on page 331. See parameters 22 and 23 below.

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 the back face.

11: The diffraction order. Multiple orders may be specified, see “Diffraction tab” on page 350.

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, set the number of X and Y facets to small numbers (1 or 2) to conserve memory and speed up ray tracing.

Note a cylinder lens results if the radius of rotation is set to zero. The reference coordinate is the center of the front face. Coating/Scatter Groups: Out edge side faces CSG #0, flat front face CSG #1, curved back face CSG #2, inside side faces CSG #3 (these are the side faces formed in the interior of an array of lenslets when decentered lens faces meet).



## Lenslet 2

A lenslet 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 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: The number of X and Y facets *for each lenslet*. See “The use of facets” on page 331. See parameters 10 and 11 below.

6: The radius of curvature of the front face of each lenslet, use zero for a plane.

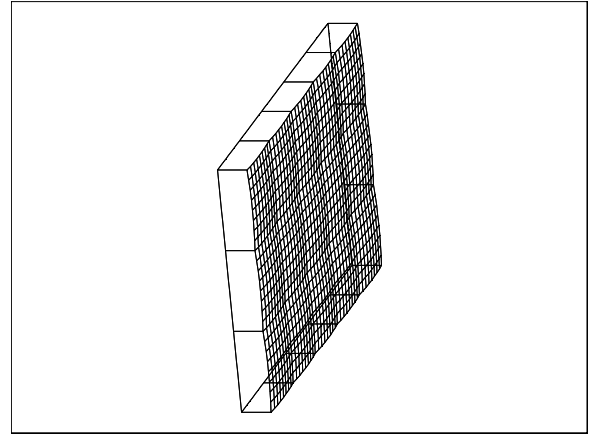
7: The conic constant of the front face of each lenslet.

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, set the number of X and Y facets to small numbers (1 or 2) to conserve memory and speed up ray tracing.

The reference coordinate is the center of the front face. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.



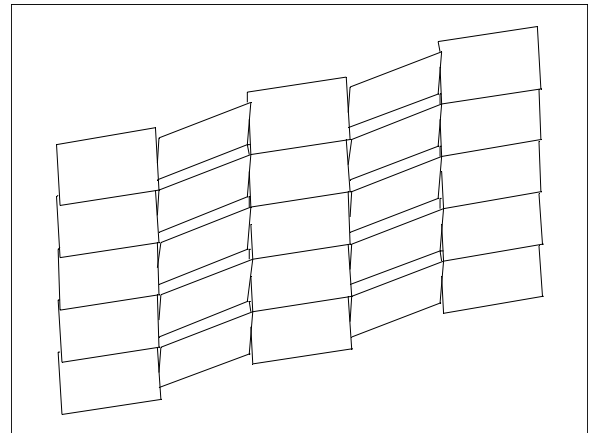
## Micro Electro Mechanical System (MEMS)

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.

Note that X for any given state of the MEMS is given by:

$$X = M1*(3^0) + M2*(3^1) + M3*(3^2) + \dots$$

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.

Coating/Scatter Groups: All faces CSG #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).

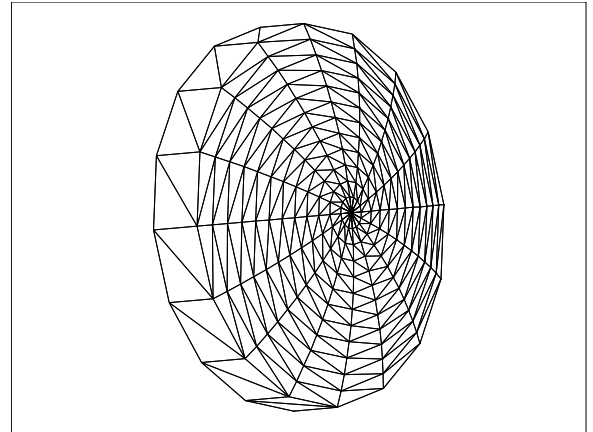
### Odd Asphere Lens

The Odd Asphere surface shape is defined by:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^{12} \alpha_i r^i,$$

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: The number of angular facets. See "The use of facets" on page 331.
- 4: The number of radial facets. See "The use of facets" on page 331.
- 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. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

## Polygon Object

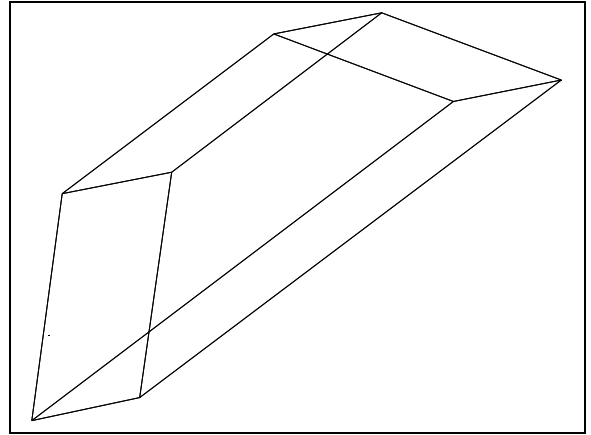
The polygon object is a very general user-defined object. It can be used to define an open polygon reflector, such as a faceted mirror; or a closed polygon with some portions reflective and others refractive or absorptive. The Polygon Object is based on a collection of 3D triangles whose vertices are placed in an ASCII file with the POB extension. See the "Defining Polygon Objects" on page 373 for more details. Any Polygon Object may be used as a detector as described in "Objects as detectors" on page 335.

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, without the POB extension. For example, if the POB file myobject.POB is placed in the \Objects directory, then specify "myobject" 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" on page 376 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" on page 217.

The reference coordinate is locally (0, 0, 0), and the polygons that compose the object may be placed anywhere relative to the reference point. Coating/Scatter Groups: Each facet belongs to a CSG as defined in the POB file, see "Defining Polygon Objects" on page 373 for details.



## Rectangular Corner

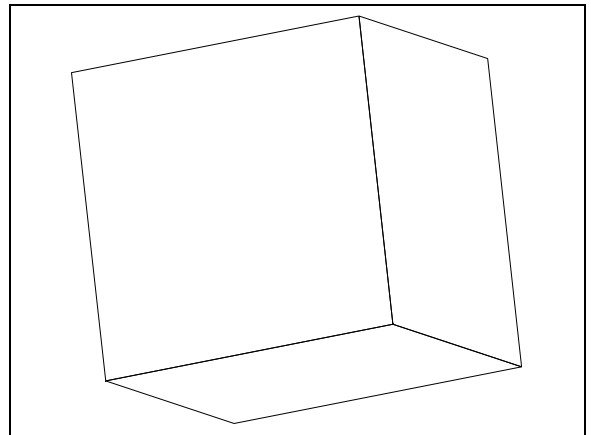
Rectangular corners are defined by 1 parameter:

- 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.

See "The use of facets" on page 331 for information on limitations of ray tracing through faceted objects.

The reference point is the point where all 3 squares touch. Coating/Scatter Groups: All faces CSG #0.



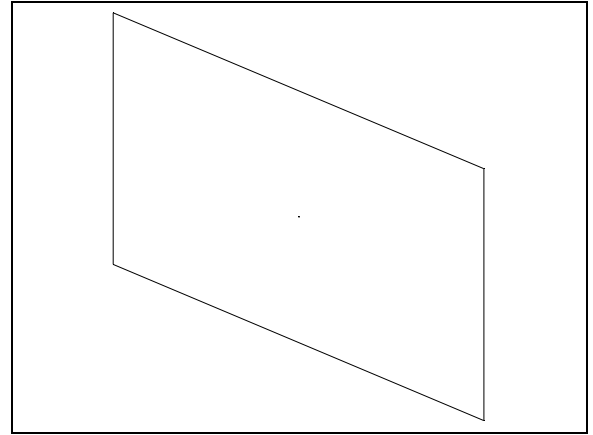
## Rectangle

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. Coating/Scatter Groups: All faces CSG #0.

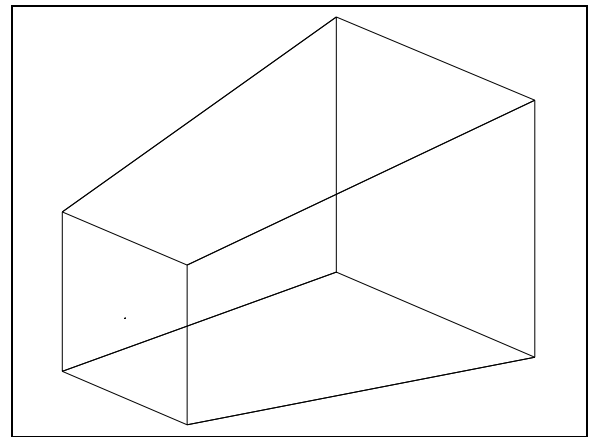


## Rectangular Pipe

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. Coating/Scatter Groups: All faces CSG #0.



## 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" on page 350.

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" on page 347.

Coating/Scatter Groups: All faces CSG #0.



## Rectangular Roof

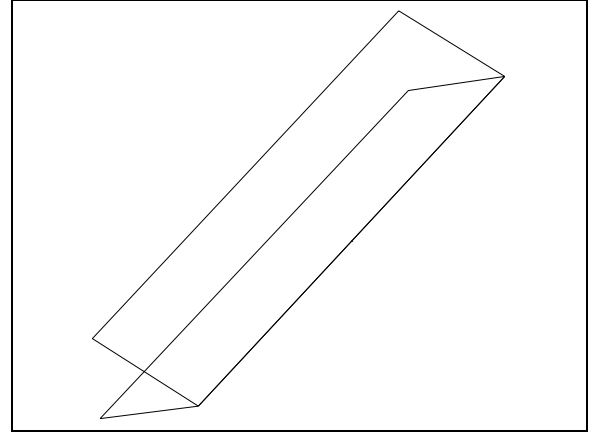
There are 3 parameters used to define a rectangular roof:

- 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.

See “The use of facets” on page 331 for information on limitations of ray tracing through faceted objects.

The reference point is the midpoint of the line joining the two rectangles. Coating/Scatter Groups: All faces CSG #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 Volume for modeling refractive solid torus shapes.

The rectangular torus surface is defined by 6 parameters:

- 1: The outer radius of the torus,  $R_{out}$ .
- 2: The inner radius of the torus,  $R_{in}$ .
- 3: The start angle of the torus,  $\theta_1$ .
- 4: The stop angle of the torus,  $\theta_2$ .
- 5: The thickness of the torus,  $T_y$ .
- 6: The # of angular facets; these divide the range  $\theta_2 - \theta_1$ . See “The use of facets” on page 331.

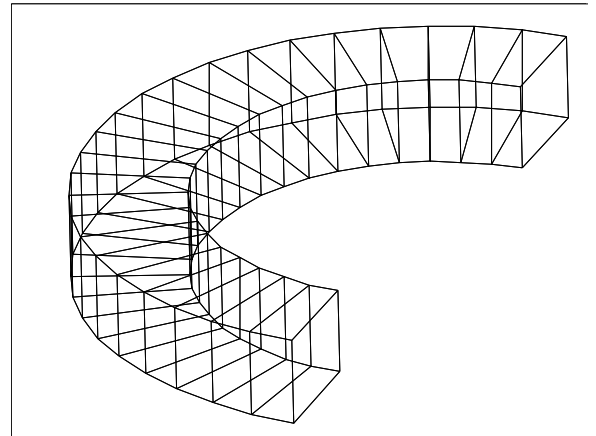
The rectangle lies in the YZ plane with the center at  $x = 0$ ,  $y = 0$ ,  $z = (R_{out} + R_{in})/2$ . This position of the rectangle corresponds to the rotation angle  $\theta = 0$ . The angles of rotation are about the Y axis and must meet this condition:

$$0 \leq \theta_1 \leq \theta_2 \leq 360.$$

There is also the restriction that  $R_{out} > R_{in} > 0$  and  $T_y > 0$ . The reference coordinate is the center of the axis of rotation. Coating/Scatter Groups: All faces CSG #0.

## Rectangular Torus Volume

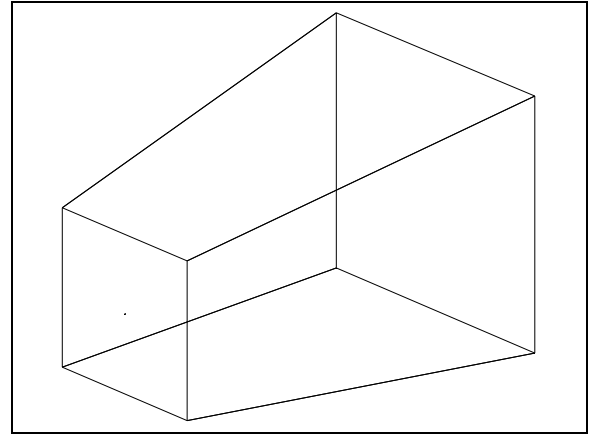
This object is essentially identical to the Rectangular Torus Surface, except the ends of the torus are closed to make a solid volume. This allows the object to be made of a refractive material. See the description of the Rectangular Torus Surface for details. Coating/Scatter Groups: Start angle end cap CSG #1, stop angle end cap CSG #2, all other faces CSG #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 6 sided solids. The reference point is the center of the front face. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

## 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” on page 350.

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” on page 347.

Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

## Slide

Slides are color RGB transparencies defined by either a BMP or JPG 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” on page 343) behind the slide, then set the slide to have a Lambertian scattering property (see “Scattering” on page 352).

Coating/Scatter Groups: All faces CSG #0.

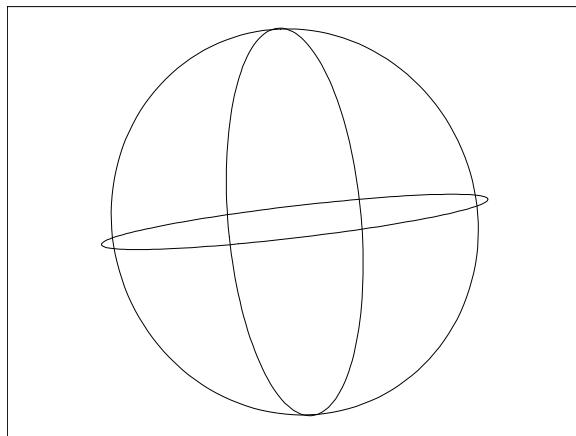
## Sphere

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.

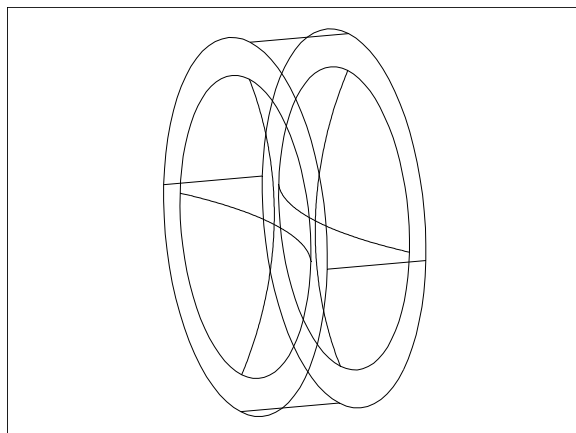
The reference point is the center of the sphere. Coating/Scatter Groups: All faces CSG #0.



## 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.

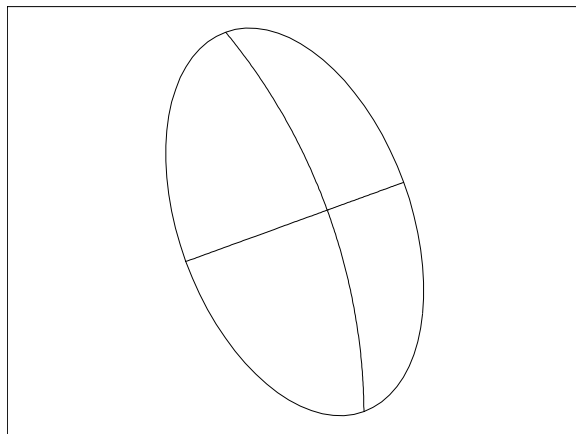
The reference point is the center of the front face of the lens. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

## 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. Coating/Scatter Groups: All faces CSG #0. This object supports user defined apertures, see "User defined apertures" on page 352. For a similar surface shape with more complex aperture shapes supported, see "Aspheric Surface 2" on page 299.

## STL Object

The STL object is a very general user-defined object. It can be used to define an open polygon reflector, such as a faceted mirror; or a closed polygon shape such as 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 ASCII 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" on page 329, "Polygon Object" on page 319, and "Imported" on page 314.

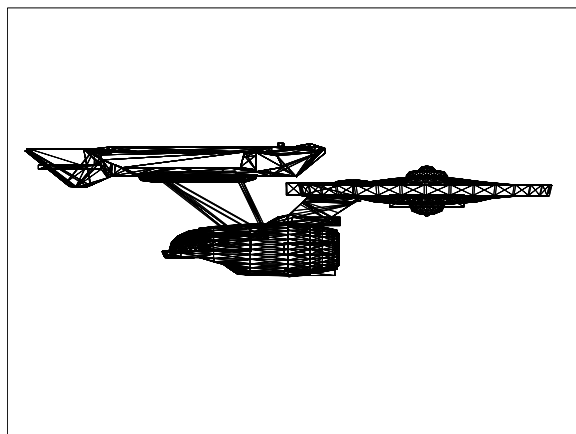
There are no fixed limits to the total number of vertices or polygons.

The STL file name is referenced in the "comments" column of the STL Object row, without the STL extension. For example, if the STL file myobject.STL is placed in the \Objects directory, then specify "myobject" 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.

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.



See "The use of facets" on page 331 for information on limitations of ray tracing through faceted objects. Coating/Scatter Groups: All faces CSG #0.

### Fast Load/Fast Trace STL objects

There are two variations of the STL object with identical parameters as described above. The Fast Load STL Object reads the triangles defined in the STL data file, and uses the raw triangle list for ray tracing and rendering. Although loading the STL object in this manner is fast, ray tracing is inefficient because of the very large number of triangles typically present in STL files. The Fast Trace STL Object reads the triangles defined in the STL data file, then sorts and pre-processes data on the triangles to make any subsequent ray tracing much faster.

The recommended usage of these two related objects is to use the Fast Load STL Object to check the accuracy, position, and scale of an STL object as placed in the system. Once the object is appropriately defined in the model, use the Fast Trace version to do extensive ray tracing. Note that if the number of triangles in the STL is small, there is no reason to not use the Fast Trace version all the time.

### Tabulated Faceted Radial

A tabulated object is based upon coordinates defined in a ASCII file ending 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 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
```

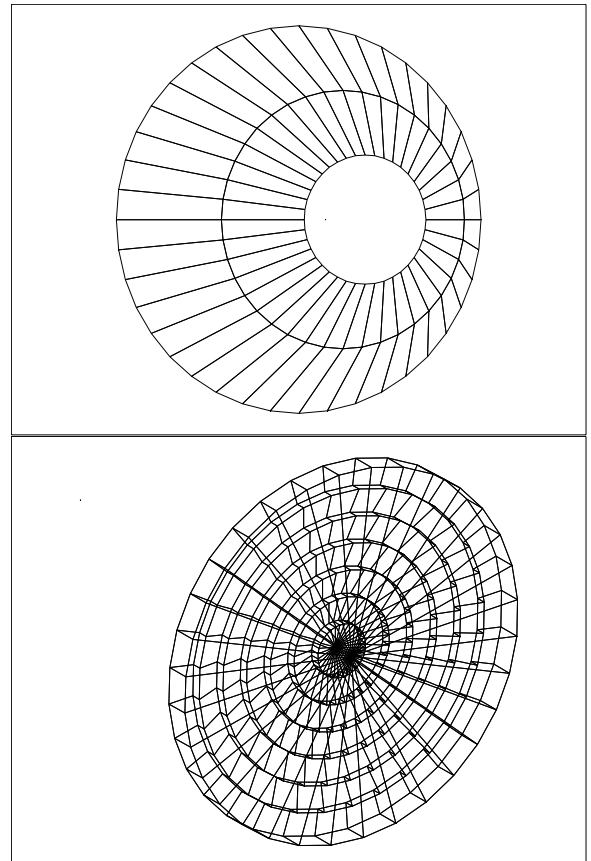
The first number of each pair is the local Y coordinate. This value must be zero or positive. The second value is the local Z coordinate. Each pair of values after the first pair represent a "zone". If there are 6 pairs of numbers, then there are 5 zones to the object. There is a maximum of 246 zones 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.

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.

See "The use of facets" on page 331 for information on limitations of ray tracing through faceted objects. Coating/Scatter Groups: All faces CSG #0.

### Tabulated Faceted Toroid

A tabulated object is based upon coordinates defined in a ASCII file ending 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.

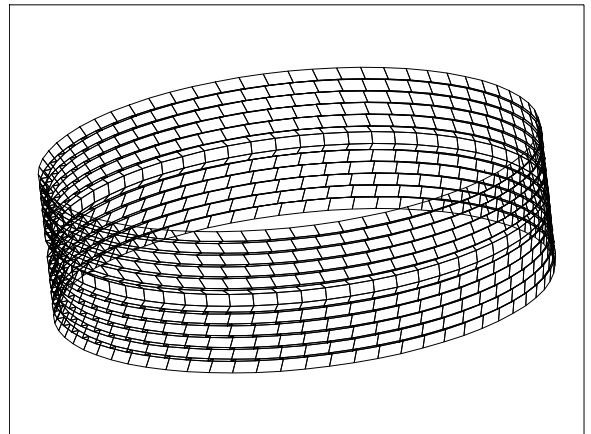
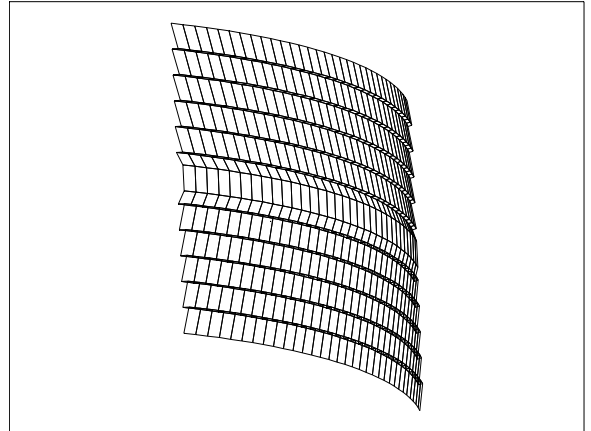
ZEMAX will generate facets which approximate a smooth surface for each zone. The facets can cover any fraction of a full circle, defined by a start and stop angle. To make a full figure of revolution, the start angle should be set to -180.0 degrees, and the stop angle to 180.0 degrees. Both angles must 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.

See "The use of facets" on page 331 for information on limitations of ray tracing through faceted objects. Coating/Scatter Groups: All faces CSG #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.

See the description of the tabulated faceted radial object for a description of this type of object and the tabulated object file format. Coating/Scatter Groups: All faces CSG #0.

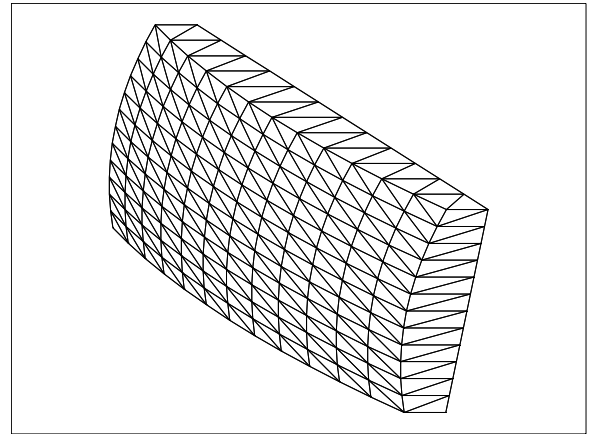
## Toroidal Lens

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:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2y^2}} + \alpha_1y^2 + \alpha_2y^4 + \alpha_3y^6 + \alpha_4y^8 + \alpha_5y^{10} + \alpha_6y^{12},$$

where c is the reciprocal of the radius of curvature in YZ plane. The toroidal lens is defined by 23 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: The number of X facets. If the lens is rotationally symmetric, then this parameter defines the number of angular facets. See "The use of facets" on page 331.



- 5: The number of Y facets. If the lens is rotationally symmetric, then this parameter defines the number of radial facets. See "The use of facets" on page 331.

- 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.

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. The reference coordinate is the center of the front face. Coating/Scatter Groups: Front face CSG #1, back face CSG #2, all other faces CSG #0.

## Toroidal Surface

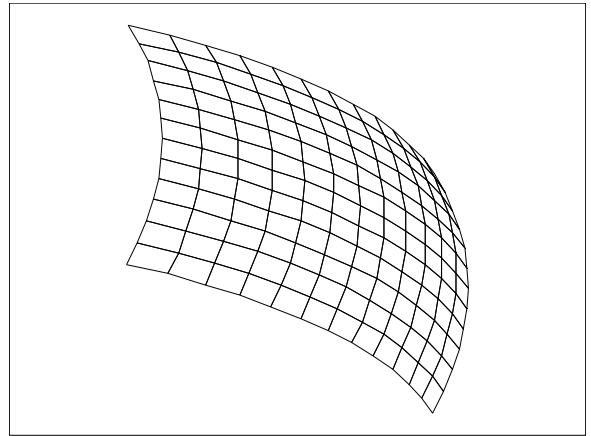
A toroidal surface consists of a rectangular surface with a possibly aspheric toroidal shape. A toroidal surface is defined by a curve in the YZ plane which is then rotated about an axis parallel to the Y axis but displaced by a distance R; the radius of rotation. The curve in the YZ plane is defined by:

$$z = \frac{cy^2}{1 + \sqrt{1 - (1+k)c^2y^2}} + \alpha_1y^2 + \alpha_2y^4 + \alpha_3y^6 + \alpha_4y^8 + \alpha_5y^{10} + \alpha_6y^{12},$$

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: The number of X facets. See “The use of facets” on page 331.
- 4: The number of Y facets. See “The use of facets” on page 331.
- 5, 6, 7: The radius of rotation, radius of curvature, and conic for the surface.
- 8-13: 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. Coating/Scatter Groups: All faces CSG #0.

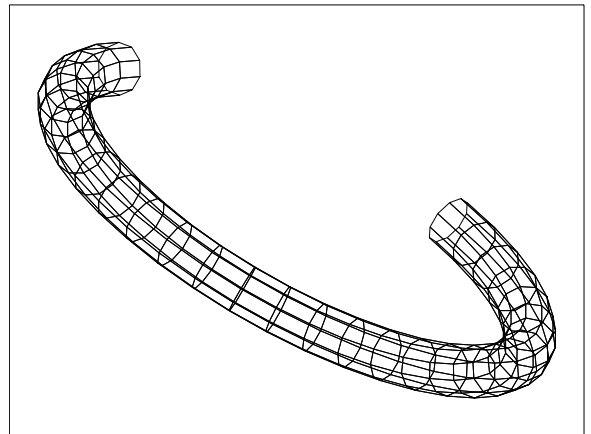


### Torus Surface

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 6 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,  $\theta_1$ .
- 4: The stop angle of the torus,  $\theta_2$ .
- 5: The # of angular facets; these divide the range  $\theta_2 - \theta_1$ . See “The use of facets” on page 331.



- 6: The # of radial facets around the circle of radius  $r$ . See “The use of facets” on page 331.

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. Coating/Scatter Groups: All faces CSG #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 be made of a refractive material. See the description of the Torus Surface for details. Coating/Scatter Groups: Start angle end cap CSG #1, stop angle end cap CSG #2, all other faces CSG #0.



## Triangular Corner

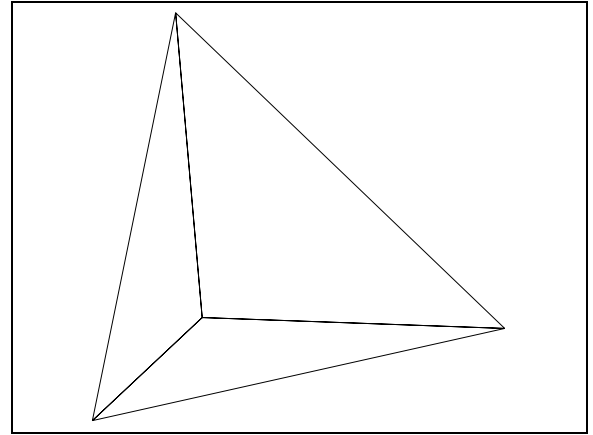
Triangular corners are defined by 1 parameter:

- 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.

See “The use of facets” on page 331 for information on limitations of ray tracing through faceted objects.

The reference point is the point where all 3 triangles touch. Coating/Scatter Groups: All faces CSG #0.

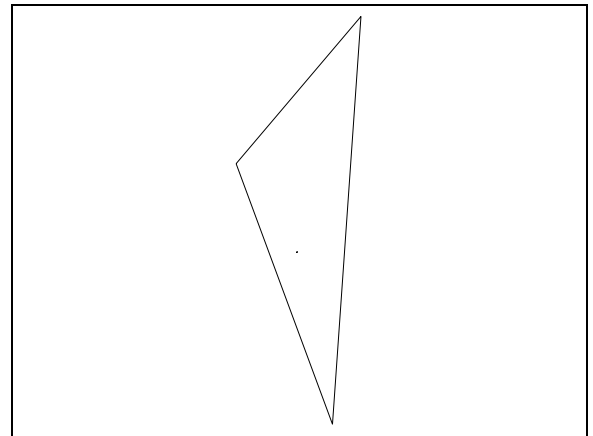


## Triangle

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. Coating/Scatter Groups: All faces CSG #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” on page 319. Objects may also be defined in an external CAD program and then imported into ZEMAX, see “Imported” on page 314 and “STL Object” on page 324. There are also methods for user definitions for gradient index media (“Defining GRIN media for non-sequential ray tracing” on page 359), surface scattering properties (“Scattering” on page 352), bulk scattering properties (“Bulk scattering” on page 357), and diffraction (“Defining DLLs for ray splitting at diffractive surfaces” on page 361).

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 coating and scatter groups.
- 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. There are four different parts to the DLL code:

- 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 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 \OBJECTS\DLL\USEROBJECTS subdirectory off the main ZEMAX directory.***

See "The use of facets" on page 331 for information on limitations of ray tracing through faceted objects. Coating/Scatter Groups: All faces have user definable CSG #s.

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

DLL name	Description
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; and selectable number of facets. This DLL demonstrates the creation of a list of triangles to approximate the object shape, associating object faces with coating and scatter groups, and exact ray-surface iteration to arbitrary surface description functions.
HalfCylinder	A half cylinder, with user definable radius, length, and number of facets. This DLL demonstrates the creation of a list of triangles to approximate the object shape, associating object faces with coating and scatter groups, and exact ray-surface iteration to arbitrary surface description functions.

## Zernike Surface

A Zernike surface is defined by the following sag equation:

$$z = \frac{cr^2}{1 + \sqrt{1 - (1 + k)c^2r^2}} + \sum_{i=1}^8 \alpha_i r^{2i} + \sum_{i=1}^N A_i Z_i(\rho, \phi),$$

where  $c$  is the curvature of the surface,  $k$  is the conic constant,  $r$  is the radial coordinate, the  $\alpha$  terms are aspheric coefficients,  $N$  is the number of Zernike coefficients in the series,  $A_i$  is the coefficient on the  $i^{th}$  Zernike Standard polynomial,  $r$  is the radial ray coordinate in lens units,  $\rho$  is the normalized radial ray coordinate, and  $\phi$  is the angular ray coordinate. The Zernike Standard polynomials are defined in the table given in "Zernike Standard Coefficients" on page 168. 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.

4: The minimum radial aperture in lens units. This value may be zero.

5: The number of angular facets. See "The use of facets" on page 331.

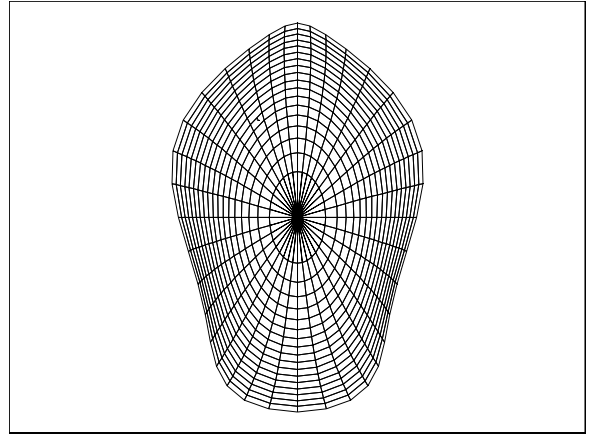
6: The number of radial facets. See "The use of facets" on page 331.

7-14: The even radial aspheric coefficients  $\alpha$ .

15: The number of terms to use in the Zernike expansion.

16-247: The Zernike coefficients.

Coating/Scatter Groups: All faces CSG #0. This object supports user defined apertures, see "User defined apertures" on page 352.



## The use of facets

Some objects support a user defined specification for the number or radial, angular, or x/y facets, such as the Biconic Lens, and many others. It is important to note that the facets are only used for two purposes:

1) To render the object

2) To provide a "first guess" as to the location of the ray-object intercept point

An arbitrary ray may intersect an object in multiple places. For accurate ray tracing, iteration is used to find the exact intercept point. Iteration requires a good close first guess; otherwise the ray may converge to an incorrect solution, or may not converge at all. For ray tracing, the faceted representation is used only to get the rays close enough to the actual surface so that iteration works efficiently. The actual surface shape of the object is modeled exactly for ray tracing purposes; the accuracy of the ray trace is not limited by the faceted approximation.

The number of facets needs only to be high enough to render the object with acceptable quality and to preclude erroneous solutions for the iterative ray intercept computation. Using more facets will cause the ray tracing to be slower and not increase the accuracy of the ray intercept positions or normals. If rays seem to be missing an object, try increasing the number of facets in the representation.

## **Detectors**

There are four types of detectors in ZEMAX summarized in the table below.

SUMMARY OF NSC SOURCES AND DETECTORS

Object Name	Description	Page
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.	332
Detector Surf	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.	334
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.	335
Faceted objects as detectors	Most faceted objects of arbitrary shape may be used as a detector that records incoherent irradiance data.	335

Detailed descriptions of these detector types follow.

### **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.
- 4) # Y Pixels: The number of pixels along the y direction.

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" on page 87.
- 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" on page 369.
- 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 an approximation to the 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.

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. 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. Setting the polarization flag to 1, 2, or 3 will also consider the phase of the electric field as well as the phase of the ray (due to optical path, index, and phase objects) when computing the coherent amplitude and phase displayed by the detector.

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" on page 369.

Five 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 part, imaginary part, and amplitude 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" on page 87.

### 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 both the amplitude and 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. ZEMAX cannot determine where the energy went, and therefore cannot account for conservation of energy in coherent irradiance calculations.

To compute the coherent irradiance from the ray data, ZEMAX first computes a coherence factor. The coherence factor is the sum of the real part of the amplitude of all rays squared, plus the sum of the imaginary part of the amplitude of all rays squared, divided by the sum of the amplitude for all rays squared. Defined this way, the coherence factor is 1.0 if all rays constructively interfere, and is zero if all rays destructively interfere. Finally, this coherence factor is multiplied by the sum of the incoherent intensity to yield the coherent intensity.

It is also 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 single point or planar collimated sources. The initial phase of the ray and the coherence length of the source may be defined, see "Coherence length modeling" on page 359 and "Sources tab" on page 350.

### 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" on page 298 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" on page 298 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 default settings for the Detector Viewer:

- 7) Data type: Select 0 for incoherent irradiance. This is currently the only valid option.
- 8) Color: Select for 0 grey scale, 1 for inverse grey scale, 2 for false color, and 3 for inverse false color.
- 9) Smoothing: The amount of smoothing. This value is not supported for drawing on shaded model plots, but is used to set the default on the Detector Viewer display. The algorithm is described in "The Detector Viewer" on page 369.
- 10) Scale: Choose 0 for linear, 1 for log -5, 2 for log -10, and 3 for log -15.
- 11) Plot 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  $\alpha$  coefficients on the polynomial expansion. See "Aspheric Surface" on page 298 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" on page 369.

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" on page 87.

### Comments about detector triangles

ZEMAX uses a collection of triangles to conform to the shape of the curved surface. There are  $N_r$  radial zones, and  $N_a$  angular zones. Together these two parameters define  $N_r * N_a$  polygonal regions. Each region is generally divided into two triangles, so the total number of triangular pixels is  $2 * N_r * N_a$ . However, if the minimum aperture (parameter 4 above) is zero, then the innermost radial zone has only  $N_a$  triangles rather than  $2 * N_a$ . In this case the total number of triangles is  $2 * N_r * N_a - N_a$ . 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.

### 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  $n_x * n_y * n_z$ .

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. 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" on page 369.

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.

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" on page 87.

### Objects as detectors

Any object type that ZEMAX internally represents as a collection of triangles may act like a detector. This includes objects that use facets to form flat faces, such as the polygon, STL, and rectangular volume objects. Objects that approximate the shape of a curved surface or solid for rendering purposes by using facets, 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 on each triangular facet used

to render the object 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 on the facet or may be normalized to be flux per unit area the facet represents.

Objects acting as detectors can only display incoherent intensity. Each individual triangle used to draw the faceted object becomes a single pixel. All rays which strike the pixel will be summed to yield the total intensity. The faceted 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" on page 217.

## **Sources**

Sources include points, ellipses, rectangles, volumes, 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).

### **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. 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 number of the object in which the source is placed must be provided on the "Inside Of" data column of the source. If this first object is in turn inside of a second object; then the first object must indicate 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.

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.

## SUMMARY OF NSC SOURCES

Object Name	Description	Page
Source Diode	An array of diodes with separate X/Y distributions	337
Source DLL	A source defined by an external user supplied program.	338
Source Ellipse	An elliptical surface that emits light from a virtual source point.	339
Source Filament	A source in the shape of a helical filament.	340
Source File	A user defined source whose rays are listed in a file.	340
Source Gaussian	A source with a Gaussian distribution.	342
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.	342
Source Radial	A radial symmetric source based upon a spline fit of arbitrary intensity vs. angle data.	342
Source Ray	A point source aligned with direction cosines.	343
Source Rectangle	A rectangular surface that emits light from a virtual source point.	343



Object Name	Description	Page
Source Tube	A source in the shape of a cylindrical tube.	343
Source Two Angle	A rectangular or elliptical surface that emits light into a cone with distinct angles in the X and Y directions.	343
Source Volume Cylinder	A volume source in the shape of a cylinder with an elliptical cross section.	343
Source Volume Ellipse	A source in the shape of an elliptical volume.	344
Source Volume Rectangle	A volume source in the shape of a rectangle.	344

The individual sources are described in the following sections.

### Parameters common to all source objects

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. See "Choosing the number of analysis rays" on page 365.
- 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" on page 87 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.

### 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  $\alpha_x$  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. If  $G_x$  is greater than 1.0, then the distribution becomes more "squared". Both  $G_x$  and  $G_y$  must be greater than or equal to 1.0.

Most laser diode manufacturers specify the far field divergence angles as the full width of the distribution between the half power points,  $\theta_{fwhm}$ . For a true Gaussian distribution ( $G_x = 1$ ), setting the left hand side of the equation to  $\frac{1}{2}I_o$ , setting  $\theta_y$  to zero, substituting for  $\theta_x$  the value of  $\frac{1}{2}\theta_{fwhm}$ , then solving for  $\alpha_x$  gives

$$\alpha_x = \frac{\theta_{fwhm}}{\sqrt{2\ln(2)}}, \text{ or}$$

$$\alpha_x = (0.8493218)\theta_{fwhm}.$$

For example, a diode with a  $\theta_{fwhm}$  in the x direction of 10 degrees, the value for  $\alpha_x$  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 and number y, along with the delta x and delta y spacings may be defined.

The parameters used to define this source are:

- 1-5) See "Parameters common to all source objects" on page 337.
- 6) Astigmatism: The distance to offset the XZ distribution in lens units.
- 7) X divergence in degrees.
- 8) X supergaussian factor.
- 9) Y divergence in degrees.
- 10) Y supergaussian factor.
- 11-12) Number of diodes in X/Y.
- 13-14) Spacing of diodes in X/Y in lens units.

## 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 on page 340.

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" on page 377.

### 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

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 \OBJECTS\DLL\SOURCES subdirectory off the main ZEMAX directory.**

### Sample Source DLLs

The following sample source DLLs and source code are included with ZEMAX.

#### SAMPLE SOURCE DLLS

DLL name	Description
FIBER1	<p>A planar fiber model. The rays emanate from a disk of radius R. The intensity profile is given by</p> $I(r) = A + Br^2 + Cr^4,$ <p>where</p> $0.0 \leq r \leq R.$ <p>Once the ray starting position is determined, the ray emanates from the surface into a cone whose numerical aperture depends upon the radial coordinate:</p> $NA(r) = D + Er^2 + Fr^4.$ <p>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.</p>

### 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 greater than or equal to unity, and need not be an integer. When used in this mode, the source acts like a diffuse cosine source. The larger Cn, the narrower the distribution becomes. Note this distribution is rotationally symmetric about the local Z axis.

- c) Rays emit in a Gaussian distribution of the form

$$I(l, m) \approx I_o e^{-(G_x l^2 + G_y m^2)},$$

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" on page 337.
- 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.

10) Gaussian Gx: The X term in the Gaussian distribution. Ignored if Cn is not zero.

11) Gaussian Gy: The Y term in the Gaussian distribution. Ignored if Cn 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.

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.

### 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" on page 337.

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.

### Source File

The Source File is a source whose ray coordinates, cosines, and intensity are defined in a user supplied file. This feature allows creation of arbitrary user defined sources. The name of the file containing the ray data must be placed in the comment column of the object. The file extension must be DAT and the file must be placed in the \OBJECTS directory. The file format may be either ASCII or binary, both formats are described below.

The parameters are:

1-5) See "Parameters common to all source objects" on page 337.

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 1,000,000 or less. Files with more than 1,000,000 rays are too large to hold in memory while randomizing.

7) Total (units): This value is provided for reference only, and is the total power in source units defined by all the rays in the file. This value is set by ZEMAX upon reading the file and should not be altered or set by the user. The actual power of each ray is defined by parameter 3 above and the number of rays being traced.

### ASCII vs. binary format files

The maximum number of rays currently allowed for ASCII files is 1,000,000. If there are more than 1,000,000 rays defined in an ASCII 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 DAT 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 ASCII file. This conversion is done to save disk space (binary files are about 30% as large as equivalent ASCII files), load time (binary files read about 20x faster than ASCII files), and system memory (large binary files are limited

only by available disk space and require very little RAM, even if billions of rays are being traced, because the data is left on disk).

### Memory requirements for source file objects

For both ASCII and binary files, if the total number of rays to be traced is less than 1,000,000, ZEMAX stores the rays in system memory for maximum speed. This requires about 28 bytes of memory for each ray, or up to 28 Mb of memory for 1,000,000 rays. If the total number of rays is 1,000,000 or greater, 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.

### Restrictions on the number of rays selected

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.

### Binary Source File format

The Binary Source File consists of a header structure of the form:

```
typedef struct
{
    int Identifier; // Will be set to 8675309 for quick check of proper format
    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 AzimuthBeg, AzimuthEnd; // Angular range for ray set (Degrees)
    float PolarBeg, PolarEnd; // 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; // Scale factor to expand/contract source
    float unused1, unused2, unused3, unused4;
    int reserved1, reserved2, reserved3, reserved4;
} NSC_RAY_DATA_HEADER;
```

The data types float and int are both 32 bit types. The "Identifier" integer must be the value 8675309. The other data may or may not be included; ZEMAX only uses the NbrRays and DimensionUnits parameters. After the header follow NbrRays ray structures. Each ray structure is of the form:

```
typedef struct
{
    float x, y, z;
    float l, m, n;
    float intensity;
} NSC_RAY_DATA;
```

The binary format described above is used by ZEMAX when generating source files from ray trace data (see the discussion of "Save Rays on Object n" in the section "Ray Database Viewer" on page 367). This format is also supported by the ProSource program available from Radiant Imaging ([www.radiimg.com](http://www.radiimg.com)). To generate a binary source data file for use in ZEMAX from the ProSource program follow these instructions:

Have a source file (\*.rsm) open in ProSource. This enables the "Gen Source" button on the main toolbar.

Click on the "Gen Source" button on the main toolbar. This opens the Ray Generation dialog box.

From the "File Format" area of this dialog box, choose "ZEMAX Binary".

Click on "Generate Rays", at the bottom, to close the dialog box and create the binary file.

These instructions may not be complete or current, and the documentation for ProSource should be consulted for more information.

### ASCII Source File format

The ASCII 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

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 ASCII 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 "ASCII vs. binary format files" on page 340.

### Intensity normalization in source files

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.

### 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" on page 337.

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.

### 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" on page 337.

6) Cone angle: The semi-cone angle in degrees.

### 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 180, inclusive. The angular range from 0.0 to 90.0 degrees 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 with a maximum possible resolution of about 0.5 degrees. 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" on page 337.

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.

8-9) Unused, reserved for future expansion of this feature.

10) The number of points, must be between 5 and 180, inclusive.

11+) The relative intensity data at each angle from the normal.

## 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" on page 337.

6-8) Cosines: The local X, Y, and X direction cosines of the ray. These are automatically normalized.

## 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.

## 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" on page 337.

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.

## 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 is uniform in X and Y, with distinct maximum half angles for each direction. The angular distribution may also be either rectangular or elliptical.

The parameters used to define this source are:

1-5) See "Parameters common to all source objects" on page 337.

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.

9) Y Half Angle in degrees: The half angle of the cone of rays in the YZ plane.

10) Spatial Shape: Use 0 for rectangular, 1 for elliptical.

11) Angular Shape: Use 0 for rectangular, 1 for elliptical.

## 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" on page 337.

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.

### Source Volume Ellipse

The Source Volume Ellipse is a 3D volume formed by an elliptical 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 ellipse satisfy this relation:

$$\left(\frac{X}{W_x}\right)^2 + \left(\frac{Y}{W_y}\right)^2 + \left(\frac{Z}{W_z}\right)^2 \leq 1,$$

where W refers to the half widths in X, Y, and Z. The parameters are otherwise identical to that for the Source Volume Cylinder.

### 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.

### 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.

### The object coordinate system

Each object's position is defined by 6 parameters: the X, Y, and Z coordinates, and the rotation about the global Z, then global Y, then global X axis. Note the coordinate translation is done first, then the tilts, in the order Z, Y, and X. This is the same convention as for the coordinate break surface when using order flag = 0 (see page 239). 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" on page 93.



## 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 matrixes then become:

$$G' = O' + R'G$$

$$' = O' + R'[O + RL]$$

$$' = [O' + R'O] + [R'R]L$$

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.

## Placing objects inside of, adjacent to, or overlapping other objects

Placing sources within other objects is described in "Placing sources inside objects" on page 336.

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

When two NSC objects are placed in contact, such as a lens touching one face of a prism, numerical round off 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 and the finite precision of the computer. Objects placed within the glue distance of one another are automatically "cemented" together for ray tracing purposes.

The glue distance is the distance below which the objects are considered in contact. This value should only need adjustment from the default value of 1E-06 lens units (or 0.001 micrometers if the lens units are millimeters) in very rare cases.

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. This can affect ray tracing results when tracing rays from faceted reflectors, and a ray strikes within the glue distance of the edge between two angled facets, see page 376 for a discussion. In most cases, no adjustment should be made to the glue distance. However, the value may be edited on the Non-Sequential tab of the System, General dialog box.

## 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 tab of the General dialog box. 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.

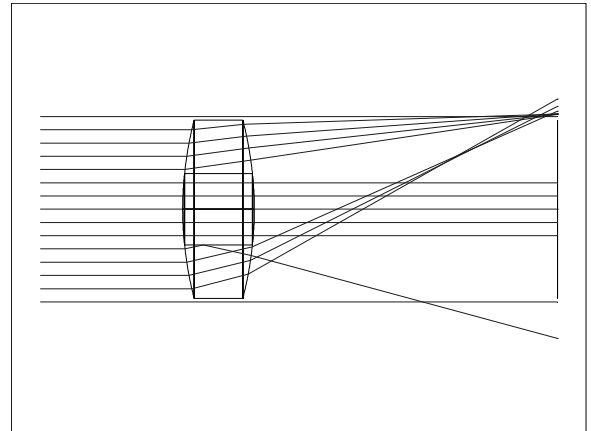


***If a ray strikes more than one object at the exact same point in space; the last object listed in the NSC Editor determines the properties of the surface or volume at that point.***

---

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" on page 359). 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", or a homogeneous glass name as defined in one of the currently selected glass catalogs. 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). The properties of objects are described by the following table.

OBJECT PROPERTIES BY MATERIAL

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. Examples include flat, curved, and faceted mirrors. 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, solid light pipes, and closed faceted volumes. Absorbers may be surfaces or volumes with the material name set to "ABSORB". Polygon Objects may be composed of facets, and the individual facets may be refractive, reflective, or absorptive. Note that STL Objects act just like other volumes, and not like Polygon Objects, because the STL format does not support individual facets having refractive, reflective, or absorptive properties.

### **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.

Ray splitting is supported on diffractive surfaces, but splitting is only allowed by order and not by reflected and transmitted Fresnel coefficients. For this reason, no "ghost" rays are generated from diffractive surfaces.

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 "Diffraction tab" on page 350.

## **The object properties dialog box**

For all objects inserted into the NSC Editor, the object properties dialog box is used to define properties such as the object type, special aperture settings, scattering and coating properties, and gradient index properties. The previous/next object buttons allow rapid navigation through all objects. The object properties dialog box has multiple tabs as described below.

### **Type tab**

The Type tab supports the following controls:

Type: Used to select the general object type, for example, sphere, ellipse, rectangle, or other type.

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. For more information see page 352.

File: If a user defined aperture is selected, the name of the aperture file may be selected from this control.

Edit User Aperture: This button will invoke a text editor to allow user editing of the selected UDA file. The UDA file needs to be saved and the object reloaded to make the changes effective.

UDA Scale: The UDA Scale is a dimensionless multiplier that scales the aperture defined in the UDA file.

Color: This control is used to select the color the object will be drawn with on the shaded model display.

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.

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 user defined colors are described in "Colors" on page 63. The coloring of rows may be disabled, see "Editors" on page 61.

Consider Objects: For a detailed discussion of this feature see "Defining paths using the Consider Objects list" which follows on page 351.

Use Consider 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. If unchecked, all possible object intersections will be checked. See "Defining paths using the Consider Objects list" which follows on page 351.

Rays Ignore This 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. Checking this option on 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.

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.

Object Is A Detector: If checked, then rays striking facets on the object will increment a detector associated with the faceted representation. Only objects that have a faceted representation support this feature; for other objects, this option will be unavailable.

Show As: This control selects the color scheme used to represent faceted objects that are detectors on the shaded model plot. Color by flux will color each facet according to the total power incident on that facet. Color by irradiance will color each facet by the power per area; which is the flux divided by the area the facet represents. Generally choosing irradiance will yield more easily interpreted results if the facets are not all the same size. Only faceted objects that are detectors use this feature.

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" on page 96. The z axis is twice this length. The vertex of the coordinate indicator is at the local vertex.

## Coating/Scattering tab

The Coating/Scattering tab supports the following controls:

Coating/Scatter Group: For a description of coating/scattering groups, see page 352.

Profile: A profile is a collection of settings related to the thin film coatings and scattering model data applied to a coating scattering group. 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; the data may be typed in once, then saved to a profile with a user defined name. Once saved, any other object may use the same profile. Profiles are stored in the file whose name is specified on the Files tab of the System, General dialog box; see "Scatter Profile" on page 91 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.

Coating: The name of the coating to apply to the surfaces within the selected group. See page 352. Coatings are defined in the system coating file, see "Defining coatings in ZEMAX" on page 501 for details.

Scatter Towards: For a detailed discussion of this feature see "Defining a Scatter Towards list" which follows on 351.

For all other scattering controls, see the discussion "Scattering" on page 352.

## Bulk Scattering tab

The Bulk Scattering tab is used to define bulk (volumetric) scattering of solids. The tab supports the following controls:

Model: Select either no bulk scattering, angle scattering, or DLL defined scattering. For a detailed description of bulk scattering, see "Bulk scattering" on page 357.

Mean path: The mean path between scatter events.

Angle: The angular cone in which the rays scatter.

DLL: The name of the DLL which defines the scattering function.

Remaining controls: The remaining controls are used to define the parameters to be passed to the DLL. To make these parameters variable or under multi-configuration control, use the NPRO operand; for details see "NPRO" on page 473.

## GRIN tab

The GRIN tab is used to define the properties of solids made of a gradient index material. The tab supports the following controls:

Use DLL defined GRIN media: If checked, then the object will use an externally provided DLL to define the properties of the gradient index media.

DLL: The name of the DLL to use.

Maximum Step Size: The maximum step size to use during piecewise ray tracing.

Remaining controls: The remaining controls are used to define the parameters to be passed to the DLL. To make these parameters variable or under multi-configuration control, use the NPRO operand; for details see "NPRO" on page 473.

For more information on defining GRIN media, see page 359.

## Diffraction tab

For important information that applies to all diffractive objects, see “Diffraction from NSC objects” on page 347.

The Diffraction tab is used to define the properties of diffractive surfaces. The tab supports the following controls:

**Split:** Selects how rays are split off from diffractive surfaces. The options are:

**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 is defined by the user in a table.

**Split by DLL function:** An external DLL program is used to define which orders are traced, how much energy each order is given, and optionally, what the output ray properties are (for user defined diffraction). For more information on defining Diffraction DLLs, see page 361.

**DLL:** The name of the DLL to use. This DLL must be placed in the \DLL\DIFFRACT subdirectory of the \OBJECTS directory. See “Directories” on page 59.

**Start/Stop order:** 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.

**Remaining controls:** The remaining controls are used to define the parameters to be passed to the DLL. To make these parameters variable or under multi-configuration control, use the NPRO operand; for details see “NPRO” on page 473.

## Sources tab

The Sources tab 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 incident polarization” on page 510. The tab supports the following controls:

**Random Polarization:** If checked, the source will emit randomly polarized light. If unchecked, the polarization state may be defined using other controls on this tab.

**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.

**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” on page 359.

**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.

**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.

Since rays in general occupy a 3D space, the electric field in the Z direction is determined by the above data assuming the  $E_x$  and  $E_y$  values are the projections down on to the local YZ plane of the source. The resulting electric field will always be perpendicular to the ray propagation vector.

The polarization properties of the source should only be set non-random if the source is reasonably well collimated. Partially polarized sources may be simulated by superimposing sources with different polarization properties at the same location.

## **Defining paths using the Consider Objects list**

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.

The Object Type tab has a user-definable list labeled "Consider Objects" for doing this. If left blank (the default), then rays leaving this object may strike any object, and ZEMAX will use it's 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.

## **Defining a Scatter Towards list**

When scattering rays for a stray light analysis, a very large number of rays may need to be traced to find a relatively small number of rays that strike an object of interest, such as a detector. The Scatter Towards feature speeds up the analysis by ignoring scattered rays which do not propagate towards an object of interest.

The Scatter Towards list is very similar to the Consider Objects list described in the preceding section. The Scatter Towards 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.

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. It is not required for an object to consider itself; ZEMAX will always check to see if a ray strikes the same object multiple times automatically.

For example, suppose object 2 defines the Scatter Towards list as "2 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 possible objects to trace towards may or may not include the object that scattered the ray.

Note that choosing an object on the Scatter Towards list does not guarantee that a scattered ray will be traced toward that object. The scattered rays are generated based upon the scattering properties of the object the incident ray strikes (see "Scattering" on page 352). ZEMAX will generate the scattered rays, then ignore those rays that do not intersect any of the listed objects. The method used by ZEMAX does not ensure that a ray will scattered toward a listed object. However, the number of scattered rays (see "Fraction to scatter and number of scatter rays" on page 352) may be made large so that it is more probable some of them will head in the desired directions.

Although algorithms exist that can guarantee that a ray scatters in a particular direction, for proper energy normalization they require that the integral of the scatter distribution function over the projected solid angle of the target object be known. Since this integral is extremely difficult to compute in virtually all cases, the algorithms require severe approximations be made to the integral which may introduce data inaccuracy. ZEMAX does not make these solid angle integral approximations, and uses only the exact scatter distribution function in computing ray energies and scatter directions.

To turn off the Scatter Towards feature, leave the list blank. In this case, all scattered rays will be traced. The key difference between the Scatter Towards list and the Consider Object list is that the Scatter Towards list only applies to scattered rays, while the Consider Object list applies to both specular and scattered rays.

## **User defined apertures**

User defined apertures (UDA) may be placed on some NSC surface objects. UDA's are defined exactly as described for sequential surfaces, see "User defined apertures and obscurations" on page 69 for a complete description. To place an aperture on an object surface, open the Object properties dialog box, choose the "Type" tab, select the "User Defined Aperture" checkbox, and select the UDA data file name from the menu. The UDA file must be placed in the \Objects directory.

## **Coating and scatter groups**

A coating/scatter group (CSG) is a collection of one or more facets or faces to an object upon which the same thin film coating and scattering properties may be applied. For example, for a singlet lens, there are 3 CSGs: 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 need to be applied; one to each coating group, if all faces of the lens are coated.

There are currently 4 CSGs per object; although most objects use only a single CSG. CSGs are numbered from 0; so the valid groups are 0 through 3. The group properties are defined with each object. Coatings and scatter properties are applied to each CSG on the Object Type selection dialog box. The CSG control also describes which CSG is used for which faces of the object selected.

## **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 tab" on page 350.

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.

See the previous section for a discussion of coating and scatter groups (CSG). Each CSG may have its own thin film coating applied.

## **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 coating and scatter group (CSG) on an object. See the section "Coating and scatter groups" on page 352 for information on CSGs. The default scatter model is "No scattering", which means no scattering will occur. The resulting ray is called the unscattered ray or sometimes the "specular" ray (even if the surface is not actually reflective).

## **Fraction to scatter and number of scatter rays**

If a scattering model other than "No scattering" or "ABg" is selected, the "Fraction to Scatter" must be defined. This fraction must be between zero (no rays will be scattered) and 1.0 (every ray will be scattered). For the ABg



model scattering, the fraction to scatter is determined by the ABg parameters, see "Defining ABg data" on page 356.

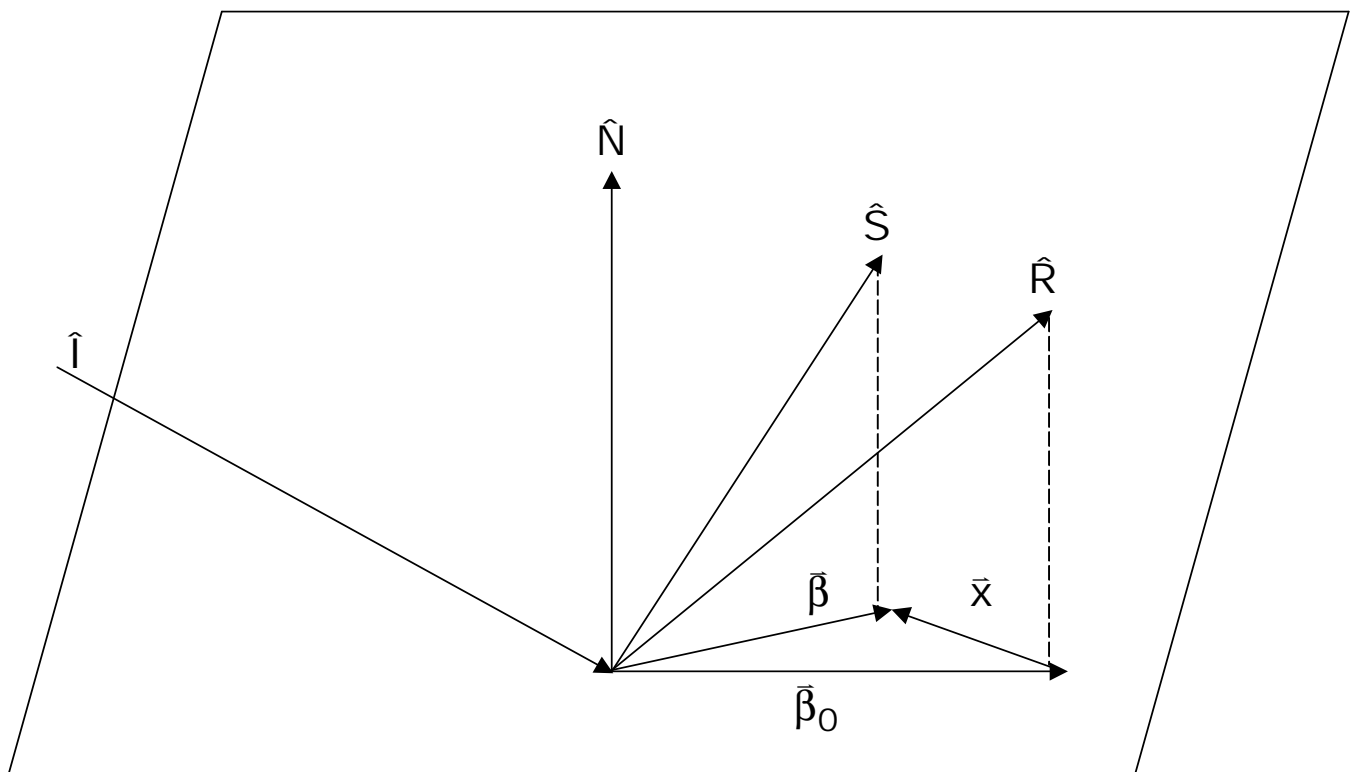
If ray splitting is off, the decision to scatter or not to scatter is made by the generation of a single random number between 0.0 and 1.0. If this random number is larger than the fraction to scatter, the ray will not scatter, otherwise, the ray will scatter. For example, if the fraction to scatter 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 fraction to scatter may be interpreted as the "Total Integrated Scatter" or TIS, which accounts for all energy other than that which is specularly reflected or refracted, or is absorbed.

### Scatter models

Scatter models are defined in terms of an intensity 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  $\hat{N}$  defines the orientation of the surface at the ray-surface interception point. The incident ray vector is  $\hat{I}$ , the specular ray vector is  $\hat{R}$ , and the scattered ray vector is  $\hat{S}$ . 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  $\hat{N}$ ,  $\hat{I}$ ,  $\hat{R}$ , and  $\hat{S}$  are all unit vectors. The projection of the specular and scattered ray vectors down to the surface are denoted by  $\vec{\beta}_0$  and  $\vec{\beta}$ , respectively. The projections are not unit vectors; the vector  $\vec{\beta}_0$  has a magnitude equal to  $\sin\theta_r$ , while  $\vec{\beta}$  has a magnitude of  $\sin\theta_s$ , where  $\theta_r$  and  $\theta_s$  denote the angle between the normal vector and the specular and scattered ray angles, respectively. The scatter vector is denoted  $\vec{x}$ , and if  $|\vec{x}|$  tends to zero, the scattered and specular vector become the same. How  $\vec{x}$  is determined depends upon the scatter model chosen.

There are five scattering models available: none, Lambertian, Gaussian, ABg, and user defined. Each available scatter model is defined in the following sections.

### No scattering

No modification to the specular ray is made,  $\vec{x}$  is zero.

### Lambertian scattering

Lambertian scatter means the scattered intensity probability distribution function goes as  $\cos\theta_s$ . Note Lambertian scattering is independent of both the ray incident and exiting angle. The scatter vector  $\vec{x}$  is chosen randomly within a unit circle, and the vector  $\vec{\beta}_0$  is set to zero length. Most diffuse surfaces are very nearly Lambertian.

### Gaussian scattering

Gaussian scattering is of the form:

$$P(\vec{x}) = Ae^{\frac{-|\vec{x}|^2}{\sigma^2}}$$

where A is a normalizing constant. A random number is used to generate an appropriate value for  $\vec{x}$ . 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 value for  $\sigma$  is defined on the CSG property dialog box, and corresponds to the 1 over e (36.79%) probability point in direction cosine space for the scatter vector magnitude. The value for  $\sigma$  should be between 1E-15 and 4.0, which correspond to nearly specular and Lambertian limits, respectively.

### ABg model scattering

The ABg scattering model is a very powerful and widely used method for defining the Bidirectional Scattering Distribution Function, or BSDF. 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  $\theta$  is measured from the normal, and  $\phi$  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 general term BSDF can refer to two separate functions, the BRDF and BTDF, for reflective and transmitted distributions, respectively.

### BSDF properties

For many optical surfaces, the BSDF is independent upon incident direction if it is plotted as a function of direction cosines instead of angles. This simplification is described in James E. Harvey, "Light-Scattering Properties of Optical Surfaces", Ph.D. Dissertation, University of Arizona (1976), and J. Harvey and A. Kotha, "Scattering Effects from Residual Optical Fabrication Errors", Proc. SPIE, July, (1999).

This representation plots BSDF as a function of  $|\vec{\beta} - \vec{\beta}_0|$ , which is  $|\vec{x}|$ . This quantity is the distance between the scattered and unscattered ray vectors when projected down to the scattering surface (see the discussion under "Scattering" above).

This scattering model is generally accepted as 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 model

When the BSDF is plotted as a function of  $|\vec{x}|$ , the resulting data may usually be fit to a function of the form

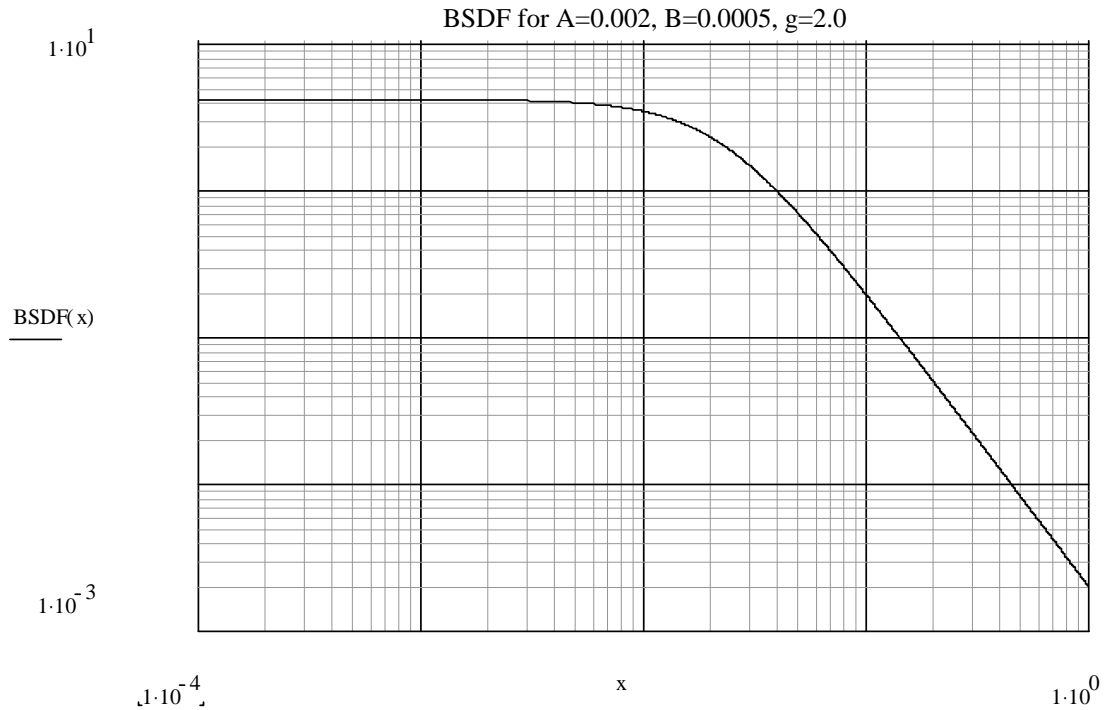
$$BSDF = \frac{A}{B + |\vec{x}|^g}.$$

The three parameters of the fit, A, B, and g, give rise to the name ABg BSDF model. For a description, see E. R. Freniere, G. G. Gregory, and R. C. Chase, "Interactive Software for Optomechanical Modeling," Proc. SPIE Vol. **3130**, p128 (1997). The following restrictions are placed on the coefficients of this model: A must be greater than or equal to zero, and B must be greater than 1E-12 unless g = 0. If g is zero, B may be zero. If A is zero, no scattering will occur. If g is zero (note g may be any value, positive or negative, but it typically is between 0 and 3), then the BSDF is constant in direction cosine space:

$$BSDF = \frac{A}{B + 1},$$

and the resulting scattering is 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  $|\vec{\beta} - \vec{\beta}_0|^g \ll B$ , the BSDF curve flattens out to a value of  $A/B$  as  $|\vec{\beta} - \vec{\beta}_0|$  goes to zero.

When  $|\vec{\beta} - \vec{\beta}_0|^g \gg B$ , 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  $x = \beta_t$ , where

$$\beta_t = \exp 10 \left( \frac{\log 10 B}{g} \right).$$

### Defining ABg data

It is frequently the case where the same ABg data will be applied to many objects in the optical system (for example, when several objects are all painted with the same type of paint). 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" on page 203. Note the ABg data defined must integrate over all possible scatter angles to a total value of less than 1.0 to conserve energy.

### Scattering rays using the ABg model

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" tab of the General dialog box, see page 91 for details.

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.

### 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" on page 377.

### 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)

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 \DLL\SurfaceScatter subdirectory of the \OBJECTS directory. See "Directories" on page 59.

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's 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.

### **Bulk scattering**

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

-DLL Defined Scattering

Each model is described below.

### 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" on page 377.

### 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 \DLL\BULKSCAT subdirectory of the \OBJECTS directory. See "Directories" on page 59.

## **Coherence length modeling**

By default, non-sequential sources in ZEMAX are either purely monochromatic, or simulated to be polychromatic 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  $\Delta x$  is the coherence length,  $c$  is the speed of light in vacuum and  $\Delta f$  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" on page 377.

### **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 \DLL\GRINS subdirectory of the \OBJECTS directory. See "Directories" on page 59.

## Sample GRIN DLLs

The following sample GRIN DLLs are provided with ZEMAX.

### SAMPLE GRIN DLLS

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.
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, <i>but not identical to</i> , 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: $n = n_0 + n_1r^2 + n_2r^4,$ where $n_x = A_x + B_x\lambda^2 + \frac{C_x}{\lambda^2} + \frac{D_x}{\lambda^4},$ for all three values of $n_x$ , which is similar to the sequential Gradient6 surface type.



DLL name	Description
GRIN9	<p>Defines a GRIN media of the form:</p> $n = n_0 \left[ 1.0 - \frac{A}{2} r^g \right], \text{ where}$ <p>where both A and <math>n_0</math> are functions of wavelength:</p> $A(\lambda) = \left[ K_0 + \frac{K_1}{\lambda^2} + \frac{K_2}{\lambda^4} \right]^2, \text{ and}$ $n_0 = B + \frac{C}{\lambda^2}.$ <p>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.</p>
GRIN10	<p>Defines a GRIN media of the form:</p> $n = n_0 + n_{y1}y_a + n_{y2}y_a^2 + n_{y3}y_a^3 + n_{y4}y_a^4 + n_{y5}y_a^5 + n_{y6}y_a^6,$ <p>where <math>y_a =  y </math> and the    symbols indicate the absolute value. This form of gradient has a discontinuity at the plane <math>y = 0</math>, and the gradient is bisymmetric about the <math>y = 0</math> plane. This is similar to the sequential Gradient10 surface type, but without the dispersion model.</p>
TORUS	<p>Defines a GRIN media of the form:</p> $n = N_0 + N_{r2}t^2,$ <p>where t is the distance from an axis of rotation defining the toroidal symmetry of the gradient index. The torus radius is one of the defining parameters. The torus radius is measured from the coordinate <math>z = 0.0</math> on the object to the location of the axis of rotation defining the torus.</p>

## **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" on page 377.

### **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 \DLL\DIFFRACT subdirectory of the \OBJECTS directory. See “Directories” on page 59.

## **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. Rays reflected from otherwise refractive interfaces are commonly called ghost reflections.

Of course, 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. Source units are defined on the Units tab of the System, General dialog box. See “Source Units” on page 87 for details.

All of these settings are defined on the Non-Sequential tab of the System, General dialog box. See “Non-Sequential” on page 94 for details.

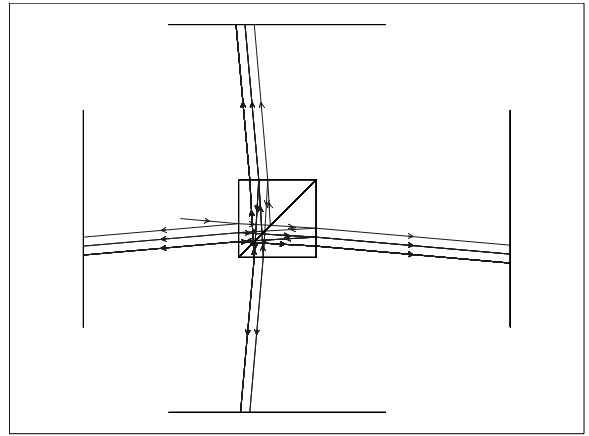
## **Ray splitting and polarization**

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 TIR's; the reflected path is of course always taken if the surface is a mirror.

The figure at right 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  $1\text{E-}8$ , 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" on page 371.

### **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:

- The object interface the ray strikes is computed first.
- If the surface is reflective or the ray TIR's, the reflected path is chosen.
- Otherwise,
- If ray splitting is on, the reflected and the refracted path are both determined.
- If ray splitting is off, the refracted path is chosen.
- The ray(s) are scattered if scattering is enabled.
- The ray(s) intensity and polarization vectors are adjusted.
- The ray(s) continue on to the next object interface.

### **Performing a Monte Carlo ray trace**

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/Detector Control**

The "Ray Trace/Detector Control" dialog box listed under "Detectors" on the NSC Editor menu bar controls the ray tracing for analysis using defined sources, objects, and detectors. The controls on the dialog box are described below.

## RAY TRACE/DETECTOR CONTROL OPTIONS

Item	Description
Clear Detectors	Select either "All" or any one specific detector to clear. This resets all of the detector's pixels to zero energy.
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.
Terminate	Stops ray tracing. This action may require a brief time.
Auto Update	If checked, all detectors will be redrawn approximately every 60 seconds, so progress may be monitored. See "Choosing the number of analysis rays" on page 365.
# 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.
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" on page 362.
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. Increasing the number of facets that represent complex objects may decrease lost ray energy. See "Lost energy" on page 365.
Save Rays	If checked, all rays traced will be saved to the file name specified. The file name only should be provided, as the path will be the same as the lens file being traced. The extension should be ZRD. The ZRD extension is used by ZEMAX for listing files in the data base viewer. Saving rays slows down ray tracing significantly, and should only be used when the data is needed for subsequent analysis. For more details, see the discussion below. 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 disk. 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 will reduce the resulting ZRD file size. For a description of the filter syntax, see "The filter string" on page 371.
Exit	Closes the dialog box.

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" on page 87). 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.

### Choosing the number of analysis rays

If the total number of rays to be traced is large, it may be important to understand the order in which the rays are traced. ZEMAX divides the ray tracing task into multiple "threads", where each thread may execute in parallel on separate CPU's if the computer is so equipped. Even on single CPU computers, the ray tracing is divided into threads. The reason ZEMAX divides the ray tracing task into threads is to give a proportional representation of all sources present early in the ray trace. If all the rays from source 1 were traced before any rays from source 2 were traced, a proportional early estimate of the final results would not be possible. The manner in which this division is done is affected by the number of analysis rays chosen for each source.

ZEMAX divides the total number of rays into an integral number of groups with equal rays, and this integral number must be a power of either 2 or 5, whichever yields the larger number of threads. Further, the number of rays in each group must be at least 400, and the total number of threads must be 1,024 or less. Both of these limits are somewhat arbitrary but are reasonable limits for efficient ray tracing.

For example, suppose a source uses 100,000 analysis rays. The power of 2 division would yield 32 threads of 3,125 rays each; note  $2^5 = 32$  and  $32 * 3125 = 100,000$ . The power of 5 division would yield 125 threads of 800 rays each; note  $5^3 = 125$  and  $125 * 800 = 100,000$ . ZEMAX would therefore choose 125 threads of 800 rays each, since this is the larger number of threads.

If multiple sources are defined, ZEMAX uses the same logic but applies it to all sources, choosing again the largest integral number of threads that evenly divides all sources. For example, suppose source 1 uses 10,000 rays and source 2 uses 4,000 rays. The power of 2 division yields 8 threads of 1,250 rays from source 1 and 500 rays from source 2. The power of 5 division yields 5 threads of 2,000 rays from source 1 and 800 rays from source 2. In this case the power of 2 division would be used because the number of threads is larger.

Once the number of threads is selected, ray tracing of the first thread begins. If additional CPU's are available for ray tracing, the second and subsequent threads are launched simultaneously as well until all CPU's are fully utilized. When each thread completes, the next thread is launched, until all threads are complete and all ray tracing is finished.

In summary, for long ray traces where early proportional results are desired, choose the number of analysis rays to be powers of 2 or 5 times 400. Large numbers that work well include 10,000, 50,000, 250,000, 500,000, 1,000,000, and 5,000,000. Larger numbers also work fine, but the total number of threads will not increase.

### 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" on page 87). 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" on page 94. 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. For most non-faceted objects, the threshold distance is *approximately* the glue distance in lens units. For facets 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" on page 95) or the object geometry needs to be checked.

### Ray database files

Ray data may be saved for subsequent analysis by checking on "Save Rays" and providing a file name in the edit box provided. The file name should end in ZRD, although any valid file name and extension will be accepted. As the rays are traced, the ray data will be saved to the file specified.

Saving the ray data will 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" on page 107 for details.

The ZRD file format is binary to keep the file as compact as possible.

The first record in the file is a single 32-bit integer indicating the version number of the ZRD file format. ZEMAX will check this number when reading the file; to be sure the format is what ZEMAX expects. 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, packed on 8 byte boundaries to yield a size of 200 bytes:

```
typedef struct
{
    unsigned int status;
    int level;
    int hit_object;
    int in_object;
    int parent;
    int storage;
    int xybin, lmbin;
    double index, starting_phase;
    double x, y, z;
    double l, m, n;
    double l2, m2, n2;
    double path_to, intensity;
    double phase_of, phase_at;
    double exr, exi, eyr, eyi, ezr, ezi;
} RAYPATH;
```

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.

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. 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 gratings 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.

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. 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 format is subject to change, however, ZEMAX will automatically convert older formats to the current format. Note the structure size is 200 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 200 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 disk.

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)
200 bytes x 5 (the segment data for ray 1)
4 bytes:7 (the number of segments in ray 2)
200 bytes x 7 (the segment data for ray 2)
4 bytes:9 (the number of segments in ray 3)
200 bytes x 9 (the segment data for ray 3)
```

For information on what ZEMAX can do with the ZRD file, see the next section.

## **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.

## RAY DATABASE VIEWER OPTIONS

Item	Description
File	The name of the ZRD file to use. ZEMAX will look in the directory 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.
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" on page 371.
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.
Save Rays on Object "n" As	<p>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" on page 340. The file name should end in the extension DAT, with no path provided. The path used will be the same as for ray source files, typically ZEMAX\OBJECTS. To prevent accidental overwrite of a previously created data file, this control is automatically "unchecked" after every save.</p> <p>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.</p>

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 Lev1 In Hit XRTS DGEF BZ X Y Z
```



0	0	0	0	0	----	----	--	0.0000E+000	5.0000E+000	0.0000E+000
1	0	1	0	2	-**-	----	--	0.0000E+000	3.0870E+000	9.0000E+000
2	1	2	0	0	*----	-*--	--	0.0000E+000	2.7044E+000	7.2002E+000
3	1	2	2	2	-**-	----	--	0.0000E+000	2.5468E+000	1.2838E+001
4	3	3	2	2	-**-	-*--	--	0.0000E+000	9.5506E-001	9.0000E+000
5	4	4	2	2	-**-	-*--	--	0.0000E+000	-6.9886E-001	1.2988E+001
6	5	5	2	2	-**-	-*--	--	0.0000E+000	-2.0350E+000	9.0000E+000
7	6	6	2	2	--*-	-*--	--	0.0000E+000	-3.2848E+000	1.2730E+001
8	7	7	0	3	*----	----	--	0.0000E+000	-5.7054E+000	1.8400E+001
9	6	6	0	0	*----	----	--	0.0000E+000	-2.9069E+000	7.3797E+000
10	5	5	0	3	*----	----	--	0.0000E+000	-4.2858E+000	1.8400E+001
11	4	4	0	0	*----	----	--	0.0000E+000	-9.6429E-002	7.4900E+000
12	3	3	0	3	*----	----	--	0.0000E+000	9.6510E-001	1.8400E+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 at the start 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.

## **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 "Performing a Monte Carlo ray trace" on page 363.

### DETECTOR VIEWER OPTIONS

Item	Description
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. Cross sections are not available in text displays.
Row/Column	Selects the row or column to display for cross section plots.

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 8 pixels. This 3 x 3 pixel 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" on page 371.
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 maximum value of the data is larger than the Max Plot Scale, the Max Plot Scale is ignored and default scaling will be applied. The Max and Min plot scale values are ignored when the Scale (see above) is Logarithmic.

### *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 half 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.

## **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
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.
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.
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. See Fn.
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.

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_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_SCATTER(n,b)	Ray segment has scattered exactly b times, and has hit object n at least once. If n is zero, any ray segment that has scattered b times will pass the test. For example, to consider only first generation scatter rays, use X_SCATTER(0, 1). See also Sn.
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 "Performing a Monte Carlo ray trace" on page 363.

### Defining Polygon Objects

Polygon objects are user-defined objects that are composed of a collection of triangles in 3D space. The objects are defined in a text file ending in the extension POB (for polygon object).

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 coating group name symbol: C

Names for the coating and scattering groups (CSG) may be defined using the C command.

#### **Syntax:**

C csgnumber "any name here"

#### **Example:**

C 0 "All faces"

The CSG names are used to display in the surface scattering tab of the object dialog box the names for the various coating and scatter groups. 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 coatinggroup

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. Note using this flag allows some triangles to be reflective, and others to be refractive or absorptive, within the same Polygon Object.

The coatinggroup is an integer defining which coating group the facet belongs to. See the section "Coating Groups" above for details. If the coatinggroup flag is omitted; coating group 0 is assumed.

#### **Example:**

```
T 1 2 3 0 2
```

This would define a refractive triangle connecting vertices 1, 2, and 3, coated with the material of coating group 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 coatinggroup
```

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.

The coatinggroup is an integer defining which coating group the facet belongs to. See the section "Coating Groups" above for details. If the coatinggroup flag is omitted; coating group 0 is assumed.

#### **Example:**

```
R 1 2 3 4 1 0
```

This would define a reflective rectangle connecting vertices 1, 2, 3, and 4, coated with the material of coating group 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 faces of the cube belong to coating group 0.

```
! A cube
! ZDC Dec 14, 1998
! 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
```

## **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 ASCII 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 \OBJECTS directory.

### **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 directory.

## **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 (see page 345) 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 DLL's 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 DLL's are quite capable of bringing ZEMAX to a crash.

*For this reason, technical support on the implementation of DLL's 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 ZEMAX Development Corporation for a quote on developing a custom DLL to meet your requirements. ZEMAX Development Corporation 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.



**Introduction**

Solves are functions which actively adjust specific values. Solves can be specified on curvatures, thicknesses, glasses, semi-diameters, conics, and parameters. Solves are set by double clicking on the cell you wish to place the solve on. The solve type control screen appears. Select the solve type from the drop-down list. Some solves require additional parameters. The available solves and required parameters are summarized in the following table. Parameters may also have "fixed" or "variable" status; these are not solves, however, they are listed for completeness. To turn solves off, select the "FIXED" option from the drop-down list. Note that this turns off the solve, it does not change the last value. Pressing Ctrl-Z twice with the cursor on the solved parameter will also remove any solve, and this is the quicker way to do so.

## SUMMARY OF SOLVES

Solve type	First parameter	Second parameter	Third parameter	Code	Integer
Curvature: Fixed					0
Curvature: Variable				V	1
Curvature: Marginal ray angle	Angle			M	2
Curvature: Chief ray angle	Angle			C	3
Curvature: Pick up	Surface	Scale factor		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
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: Pick up	Surface	Scale factor	Offset	P	5
Thickness: Optical path difference	OPD	Pupil zone		O	6

Solve type	First parameter	Second parameter	Third parameter	Code	Integer
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
Glass: Fixed					0
Glass: Model	Index Nd	Abbe Vd	Dpgf		1
Glass: Pick up	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: Pick up	Surface			P	2
Semi-Diameter: Maximum				M	3
Conic: Fixed					0
Conic: Variable				V	1
Conic: Pick up	Surface	Scale factor		P	2
Parameter: Fixed					0
Parameter: Variable				V	1
Parameter: Pick up	Surface	Scale factor	Offset	P	2
Parameter: Chief ray	Field	Wavelength		C	3

### **Curvature: 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 plane). See the F number solve.

### **Curvature: 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.

### **Curvature: Pick up**

Another useful curvature solve is the 'pick up' solve. Specifying a surface number tells ZEMAX to insert at this surface the value from some other surface. An optional scale factor allows the value to be modified. This is useful for example in systems where the light passes through the same optics twice, such as a null lens. This pickup feature allows variables to be coupled together, and they will change under the influence of other solves, editing, or optimization, in accordance with the hierarchy described below.

Pickup solves can only be used when picking up data from previous surfaces; picking up data from surfaces which follow the solve would lead to unpredictable results. The letter "P" will appear in the curvature cell to indicate that a pickup solve is active on that surface.

### **Curvature: 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.

### **Curvature: 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.

### **Curvature: 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.

### **Curvature: Element power**

The power of an element is given by:

$$\phi = c_1(n_2 - n_1) + c_2(n_3 - n_2) - c_1(n_2 - n_1) \cdot c_2(n_3 - n_2) \cdot \frac{t_2}{n_2}.$$

This solve adjusts the value of c2, 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 n3 and n2 are identical at the primary wavelength.

### **Curvature: Concentric with surface**

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.

### **Curvature: Concentric with 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.

### **Curvature: 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.

### **Thickness: Marginal ray height**

The most common thickness solve is the marginal ray height solve, which can be used to constrain the image plane to the paraxial focus. To move the image plane to the paraxial focus, double click on the thickness of the last surface before the image surface, and select "Marginal Ray Height" from the drop-down list which appears. Since the marginal ray has a height of zero (it crosses the axis) at the paraxial image plane (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 screen, and the solve immediately adjusts 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 plane. 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.

### **Thickness: 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.

### **Thickness: 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 the chapter "Conventions and Definitions" for an important discussion on the definition of edge thickness.

### **Thickness: Pick up**

Pickup solves also work with thicknesses. The pickup thickness is given by:

$$T = O + S \cdot T$$

where T is the thickness of the surface to pick up from, S is the scale factor, and O is the offset. See the "Curvature: Pick up" section for more information on pickup solves.

### **Thickness: 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 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 plane. 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. More complicated OPD constraints are available by using optimization, described in a separate chapter. The letter "O" will appear in the thickness cell to indicate that an OPD solve is active on that surface.

### **Thickness: Position**

The position solve maintains the 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.

### **Thickness: 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.

### **Thickness: 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.

### **Glass: Model**

The model glass state is not a solve. The model glass is used to idealize a glass by describing the dispersion in the visible wavelength region using three parameters: the index of refraction at d-light, the Abbe number at d-light, and a term describing the partial dispersion. For more information on model glasses, see the Chapter "Using Glass Catalogs".

### **Glass: Pick up**

Pickup solves also work with glasses, see the "Curvature: Pick up" section for a complete description.

### **Glass: Substitute**

The substitute state is not a solve. If the glass solve type is set to "Substitute", then the global optimization algorithms are permitted to change the glass type during optimization. This is similar to a variable status in that the glass type can change.

If no catalog name is given (i.e. the catalog field is blank) then glasses may be selected from all catalogs selected for use on the General System Data dialog box. 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 the Chapter "Using Glass Catalogs" for details.

See the Chapter "Global Optimization" for more details on glass substitution.

### **Glass: 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 several 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  $N_d$  and  $V_d$  as compared to the offset  $N_d$  and  $V_d$  values. See the discussion of the "Model" glass in this chapter 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: Pick up**

Pickup solves also work with semi-diameters, see the "Curvature: Pick up" section for a complete description.

## **Semi-Diameter: 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.

## **Conic: Pick up**

Pickup solves also work with conics, see the "Curvature: Pick up" section for a complete description.

## **Parameter: Pick up**

Pickup solves also work with all 8 parameters. The pickup parameter is given by:

$$P' = O + S \cdot P$$

where P is the parameter value of the surface to pick up from, S is the scale factor, and O is the offset. See the "Curvature: Pick up" section for more information on pickup solves.

## **Parameter: 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 checked on the Analysis, Calculations, Ray Trace feature; 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" on page 88) 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.

## **Suggestions for use**

There is a hierarchy by which solves are computed. The solves are computed sequentially from the first surface to the image surface, in this order: curvature, thickness, glass, semi-diameter, conic, and then parameters.

Because curvature and thickness solves can affect the entrance pupil position, ZEMAX does not allow solves which rely upon ray tracing, such as marginal ray height 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.

Solves are highly efficient, and should be used when possible instead of variables during optimization. For example, it is usually better to place a curvature solve on the last surface to control effective focal length than to optimize for focal length explicitly.



## **Introduction**

The 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 an actively damped least squares method. The algorithm is 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 the section "Modifying the merit function".

Optimization requires three steps: 1) a reasonable system which can be traced, 2) specification of the variables, and 3) specification of the 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. To reach the optimization screen, select Tools, Optimization. You must specify all variables before using optimization.

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 the chapter "Global Optimization".

## **Selecting 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. Note that Ctrl-Z is a toggle. The Multi-Configuration and Extra Data Editors also contain 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 "Optimizing glass selection" on page 424.

## **Defining the default merit function**

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. You do not have to use the default merit function, you may construct your own as described in a later section.

The easiest way to define a merit function is to select the Tools, Default Merit Function option on the Merit Function Editor menu bar. A dialog box will appear which allows selection of options for the default merit function. Each option is explained in the following paragraphs.

## **Selecting the type of optimization**

Several different types of merit functions are available. The default merit function is constructed using four key choices: The optimization type, data type, reference point, and integration method. The choices are described in the following tables.

## DEFAULT OPTIMIZATION TYPES

Name	Description
RMS	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	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 Peak-To-Valley (PTV) may be a better indicator of performance. This merit function type attempts to minimize the PTV extent of the errors.

## DEFAULT OPTIMIZATION DATA

Name	Description
Wavefront	Wavefront is the aberration measured in waves. See "Comments about RMS wavefront computations" on page 133.
Spot Radius	The radial extent of the transverse ray aberrations in the image plane.
Spot X	The x extent of the transverse ray aberrations in the image plane.
Spot Y	The y extent of the transverse ray aberrations in the image plane.
Spot X and Y	Both the x and y extent of the transverse ray aberrations in the image plane. 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.

## DEFAULT OPTIMIZATION REFERENCE POINTS

Name	Description
Centroid	The RMS or PTV computation of the data is referenced to the centroid of all the data coming from that field point. <b>Centroid reference is generally preferred, especially for wavefront optimization.</b> For wavefront optimization, reference to the centroid subtracts out piston, x-tilt, and y-tilt of the wavefront, 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. Historically, the chief ray has been used because it is simpler to compute, but this complexity is dealt with by ZEMAX with virtually no performance loss.
Chief	The RMS or PTV computation of the "data" is referenced to the chief ray at the primary wavelength.
Mean	This option is only available if the selected optimization data is wavefront. The mean reference is very similar to the centroid reference, except only the piston (mean wavefront) is subtracted out; not the x- or y-tilt. Since the exact point at which the OPD is defined to be zero is arbitrary; referencing to the mean is generally preferred to the chief ray reference for those cases where the centroid is not preferred.

### Physically significant merit functions

Note that the numerical value of the merit function is physically significant. 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.

Novice designers often ask why the RMS spot radius merit function should 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 reoptimize 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.

## **Selecting the pupil integration method**

There are two different pupil integration methods used to construct the merit function: Gaussian quadrature (GQ) or rectangular array (RA). The GQ algorithm is VASTLY superior for almost all cases of practical interest. The GQ algorithm uses a carefully selected and weighted ray set to accurately compute the RMS or PTV error over the 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 data dialog boxes, any pupil apodization function, 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 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 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, 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 GQ algorithm requires specification of the number of "Rings" and the number of "Arms", and these terms are defined in subsequent paragraphs. The only drawback to GQ is that it does not work if there are surface apertures in the optical system. For these systems, GQ is probably no longer a better choice than RA. GQ does work fine when used with vignetting factors, since the ray pattern is redistributed.

The RA algorithm traces a grid of rays through the pupil. The "Grid" size determines the number of rays traced, and is described in a subsequent paragraph. The "Delete Vignetted" option (also described later) allows the vignetted rays to be deleted from the ray set. Vignetted rays in this context are those rays clipped by surface apertures, not rays which have been altered by the use of vignetting factors (see the chapter "Conventions and Definitions"). 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.

## **Rings**

The "Rings" setting is only used in the GQ algorithm. It 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, and four rings 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 is also only used in the GQ algorithm. It 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.

You should select the number of rings and the number of arms according to the order of aberrations present in your system. 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.***

---

## Grid

The "Grid" is only used by the RA algorithm, and the value 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

The "Delete Vignetted" checkbox option is only used by the RA algorithm. If selected, then each ray in the merit function will be traced through the system, and if it is vignetted by a surface aperture, if it misses any surface, or if it is total internal reflected at any surface, 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.

## Setting thickness boundary values

Boundary constraints may be automatically generated and included in the default merit function by checking the air and/or glass boundary values on. 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. 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.

### **Start At**

The "Start At" option is used to add the default merit function at a specific position within the Merit Function Editor operand list. ZEMAX will attempt to place the default merit function after the existing targets. The algorithm used to determine the starting pointing may be ineffective if the default merit function has been edited. To control where the default start at value will be, see the DFMS operand definition.

### **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.

### **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 affect 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.

### **Pitfalls with the default merit function**

The default merit function is easy to set up, 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.



## **Optimization with apodized beams**

If no pupil apodization has been specified (see the "System" chapter for details on specifying the pupil apodization) 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. See the chapter "Conventions and Definitions" for more information about apodization.

## **Modifying the merit function**

The merit function can be modified by the user. To change the merit function, select Editors, Merit Function from the main menu bar. 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 selecting Tools, Update.

Operands are set by typing the name in the first column and then filling in the remaining data fields. There are eight fields that may be required to define an operand: Int1, Int2, Hx, Hy, Px, Py, target, and weight. The Int values are integer parameters whose meaning depends upon the operand selected. Usually, Int1 is the surface indicator, and Int2 is the wavelength indicator, but not always. Not all of the operands use all of the fields provided.

### **Int1 and Int2**

For those operands that use Int1 to indicate the surface number, the parameter specifies at which surface the target should be evaluated. Similarly, the Int2 value, when used as a wavelength specifier, describes which wavelength to use. Int2 must be an integer value equal to the wavelength number. The Int1 and Int2 parameters have other uses as described later.

### **Hx, Hy, Px, Py**

Many of the operands use Hx, Hy, Px, and Py; these are the normalized field and pupil coordinates (see "Normalized field and pupil coordinates" in the chapter "Conventions and definitions"). 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. Some operands use the Hx, Hy, Px, and Py data fields for other values, depending upon the operand type.

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.

### **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" on page 195) sums the user defined and default operands separately.

### Optimization operands

The following tables describe the available operands. The first table is a "quick reference" guide which categorizes the operands by general subject. The second table provides a detailed description of each operand (listed alphabetically) and states which operands use which data fields. Note in particular that some of the operands (such as SUMM) use the Int1 and Int2 values to indicate something other than surface and wavelength. If an operand does not use a data field, a "-" is shown.

#### OPTIMIZATION OPERANDS BY CATEGORY

Category	Related Operands
First-order optical properties	AMAG, ENPP, EFFL, EFLX, EFLY, EPDI, EXPP, ISFN, LINV, OBSN, PIMH, PMAG, POWR, SFNO, TFNO, WFNO
Aberrations	ANAR, ASTI, AXCL, BIOC, BIOD, BSER, COMA, DIMX, DISC, DISG, DIST, FCGS, FCGT, FCUR, LACL, LONA, OPDC, OPDM, OPDX, OSCD, PETC, PETZ, RSCE, RSCH, RSRE, RSRH, RWCE, RWCH, RWRE, RWRH, SPCH, SPHA, TRAC, TRAD, TRAE, TRAI, TRAR, TRAX, TRAY, TRCX, TRCY, ZERN
MTF data	GMTA, GMTS, GMTT MSWA, MSWS, MSWT MTFA, MTFS, MTFT
Encircled energy	DENC, DENF, GENC, GENF, XENC, XENF

Category	Related Operands
Constraints on lens data	COGT, COLT, COVA CTGT, CTLT, CTVA CVGT, CVLT, CVVA DMGT, DMLT, DMVA ETGT, ETLT, ETVA MNCA, MNCG, MNCT, MNCV MNEA, MNEG, MNET MNPD MXCA, MXCG, MXCT, MXCV MXEA, MXEG, MXET MNSD, MXSD TTGT, TTHI, TTLT, TTVA XNEA, XNET, XNEG, XxEA, XNEG, XXET, ZTHI
Constraints on lens properties	CVOL, MNDT, MXDT, SAGX, SAGY, SSAG, TMAS, TOTR, VOLU, NORX, NORY, NORZ, NORD
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	CENX, CENY, DXDX, DXDY, DYDX, DYDY, HHCN, IMAE, 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
General math operands	ABSO, ACOS, ASIN, ATAN, CONS, COSI, DIFF, DIVB, DIVI, EQUA, LOGE, LOGT, MAXX, MINN, OPGT, OPLT, OPVA, OSUM, PROB, PROD, QSUM, SQRT, SUMM, SINE, TANG
Multi-configuration (zoom) data	CONF, MCOL, MCOG, MCOV, ZTHI
Gaussian beam data	GBPD, GBPP, GBPR, GBPS, GBPW GBSD, GBSP, GBSR, GBSS, GBSW
Gradient index control operands	DLTN, GRMN, GRMX InGT, InLT, InVA LPTD
Foucalt analysis	FOUC
Ghost focus control	GPIM
Fiber coupling operands	FICL, FICP, POPD
Relative illumination operand	RELI, EFNO



Category	Related Operands
Optimization with ZPL macros	ZPLM
User defined operands	UDOP
Merit function control operands	BLNK, DMFS, ENDX, GOTO, OFF, SKIN, SKIS, USYM
Constraints on non-sequential object data.	NPGT, NPLT, NPVA NPXG, NPXL, NPXV NPYG, NPYL, NPYV NPZG, NPZL, NPZV NTXG, NTXL, NTXV NTYG, NTYL, NTYV NTZG, NTZL, NTZV
Non-sequential ray tracing and detector operands.	NSDC, NSDD, NSRA, NSTR
Constraints on construction optics for optically fabricated holograms	CMFV
Constraints on optical coatings, polarization ray trace data	CMGT, CMLT, CMVA, CODA
Physical Optics Propagation (POP) results	POPD
Tolerance sensitivity data	TOLR
Thermal Coefficient of Expansion data	TCGT, TCLT, TCVA

### OPTIMIZATION OPERANDS AND DATA FIELD USAGE

NAME	Description	Int1	Int2	Hxy, Pxy
ABSO	Absolute value.	Op #	-	-
ACOS	Arccosine of the value of the specified operand number. If flag is 0, then the units are radians, otherwise, degrees.	Op #	Flag	-
AMAG	Angular magnification. This is the ratio of the image to object space paraxial chief ray angles. Not valid for non-paraxial systems.	-	Wave	-
ANAR	Angular aberration radius measured in the image plane with respect to the primary wavelength chief ray. This quantity is defined as $1 - \cos \theta$ , where $\theta$ is the angle between the chief ray and the ray being traced. See TRAR.	-	Wave	Yes
ASIN	Arcsine of the value of the specified operand number. If flag is 0, then the units are radians, otherwise, degrees.	Op #	Flag	-
ASTI	Astigmatism in waves contributed by the specified surface. If the surface value 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.	Surf	Wave	-
ATAN	Arctangent of the value of the specified operand number. If flag is 0, then the units are radians, otherwise, degrees.	Op #	Flag	-

NAME	Description	Int1	Int2	Hxy, Pxy
AXCL	Axial color in lens units. This is the focal plane separation between the two most extreme wavelengths defined. The distance is measured along the Z axis. If the pupil zone is zero, paraxial rays are used to determine the paraxial focal plane locations. If the pupil zone is greater than 0.0 and less than or equal to 1.0, real marginal rays are used to determine the focal plane locations. Not valid for non-paraxial systems.	Wave1	Wave2	Pupil Zone
BIOC	Biocular Convergence. Returns the convergence between two eye configurations in milliradians. The left and right eye configurations are selected using the Int1 and Int2 values. The other parameters are: Wave: The wavelength number to use. Useaa: Use angles, if 0 field units are degrees otherwise radians. 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 "Dipvergence/Convergence" on page 154 for more information and important assumptions.	Left Eye Config	Right Eye Config	(see left)
BIOD	Biocular Dipvergence. Returns the dipvergence between two eye configurations in milliradians. See BIOC above for details.	Left Eye Config	Right Eye Config	(see left)
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.	-	-	-
BSER	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.	-	Wave	-
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. Surf is the surface number, use 0 for the image surface. Wave is 0 for polychromatic, otherwise use the monochromatic wavelength number. Hx is used to set the integer field number, Hy is 0 to ignore polarization and 1 to consider it. Px is 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.	Surf	Wave	(see left)
CENY	Centroid Y position. See CENX.	Surf	Wave	(see left)

NAME	Description	Int1	Int2	Hxy, Pxy
CMFV	<p>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 Con# is either 1 or 2, for the first or second construction system, respectively. The Opr# 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 Con# is 2 and Opr# is 7, CMFV will return the value of merit function operand 7 in construction file 2.</p> <p>If there are more than one optically fabricated hologram surfaces in the playback system being optimized, the Con# 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 Con# of 7 would indicate construction system 1 on the fourth optically fabricated hologram surface present.</p>	Con#	Opr#	-
CMGT	Boundary operand that constrains the coating multiplier of a surface or all surfaces to be greater than the target value. Use Surf = 0 for all surfaces, Layr = 0 for all layers.	Surf	Layr	-
CMLT	Boundary operand that constrains the coating multiplier of a surface or all surfaces to be less than the target value. Use Surf = 0 for all surfaces, Layr = 0 for all layers.	Surf	Layr	-
CMVA	Boundary operand that constrains the coating multiplier of a surface to be equal to the specified target value.	Surf	Layr	-
CODA	<p>Coating Data. This feature traces a polarized ray using the system global polarization state (see "Polarization" on page 90). 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.</p> <p>The polarized ray may be traced from any field point in object space to any point in the pupil, to the specified surface. If the surface is zero, the ray is traced to the image surface. Use Hx to select the field#, Hy to select the data#, and Px and Py to define the pupil coordinate. The absolute value of the data number determines what data is returned as follows:</p> <p>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</p> <p>If the data is related to the coating (1-7), and the data number is negative, the data is for the "S" polarization, otherwise the data is for the "P" polarization.</p> <p>Other coating and polarization data may be added to this list upon request.</p>	Surf	Wave	(see left)
COGT	Boundary operand that constrains the conic of surface "Surf" to be greater than the specified target value.	Surf	-	-
COLT	Boundary operand that constrains the conic of surface "Surf" to be less than the specified target value.	Surf	-	-

NAME	Description	Int1	Int2	Hxy, Pxy
COMA	Coma in waves contributed by the specified surface. If the surface value 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.	Surf	Wave	-
CONF	Configuration. This operand is used to change the configuration number during merit function evaluation, which permits optimization across multiple configurations. This operand does not use the target or weight columns.	New #	-	-
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 specified operand number. If flag is 0, then the units are radians, otherwise, degrees.	Op #	Flag	-
COVA	Conic value. Returns the conic constant of a surface.	Surf	-	-
CTGT	Center thickness greater than. This boundary operand constrains the center thickness of surface "Surf" to be greater than the specified target value. See also MNCT.	Surf	-	-
CTLT	Center thickness less than. This boundary operand constrains the center thickness of surface "Surf" to be less than the specified target value. See also MXCT.	Surf	-	-
CTVA	Center thickness value. Constrains the center thickness of surface "Surf" to be equal to the specified target value.	Surf	-	-
CVGT	Curvature greater than. This boundary operand constrains the curvature of surface "Surf" to be greater than the target value.	Surf	-	-
CVLT	Curvature less than. This boundary operand constrains the curvature of surface "Surf" to be less than the target value.	Surf	-	-
CVOL	Cylinder volume. This operand computes the volume in cubic lens units of the smallest cylinder that will contain the specified range of surfaces. 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.	First surf	Last surf	-
CVVA	Curvature value. This operand constrains the curvature of surface "Surf" to be equal to the specified target value.	Surf	-	-
DENC	Diffraction Encircled Energy (distance). This operand computes the distance in micrometers to the specified fraction of diffraction encircled, ensquared, x only, or y only (enslitted) energy. Int1 specifies the pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. Int2 is the integer wavelength number; 0 for polychromatic. Hx specifies the field number. Hy is the fraction of energy desired, and must be between zero and 1. Px is the type: 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. Py is the reference point: 0 for chief ray, 1 for centroid, 2 for vertex. If the sampling is too low, the radius returned is a 1e+10. See also DENF, GENC and XENC.	Sam- pling	Wave	(see left)

NAME	Description	Int1	Int2	Hxy, Pxy
DENF	<p>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.</p> <p>The options and settings are identical to DENC, except Hy, which here is used as the distance at which the fraction of energy is desired. See also DENC, GENC, GENF, and XENC.</p>	Sam- pling	Wave	(see left)
DIFF	Difference of two operands (OP#1 - OP#2). The two arguments are the row numbers of the operands to subtract.	Op #1	Op #2	-
DIMX	<p>Distortion maximum. This is similar to DIST, except it specifies only an upper bound for the absolute value of the distortion. The Field integer can be zero, which specifies the maximum field coordinate be used, or any valid field number. Note the maximum distortion does not always occur at the maximum field coordinate.</p> <p>If the absolute flag is 0, the value returned is in units of percentage. If the absolute flag is 1, the distortion is given as an absolute length rather than a percentage.</p> <p>This operand may not be valid for non-rotationally symmetric systems.</p>	Field	Wave	(see left)
DISC	<p>Distortion, calibrated. This operand computes the calibrated distortion across the field of view, and returns the absolute value of the maximum deviation from linearity of the f-theta condition.</p> <p>If the absolute flag is 0, the value returned is in units of percentage. If the absolute flag is 1, the distortion is given as an absolute length rather than a percentage.</p> <p>This operand is extremely useful for designing f-theta lenses.</p>	-	Wave	(see left)
DISG	<p>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 any wavelength, using any field point as a reference. The method used and assumptions made are the same as for the grid distortion plot, see “Grid Distortion” on page 158. 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.</p> <p>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</p> $H = \frac{\tan(\theta)}{\tan(\theta_M)}$ <p>where <math>\theta</math> is the angle relative to the reference field chief ray and <math>\theta_M</math> is the maximum field angle (see “Maximum field” on page 48). These definitions are needed to support the reference field feature.</p> <p>If the wavelength is a positive number, DISG returns the distortion as a percentage. If the wavelength is a negative number, the absolute value of the wavelength number is used to define the wavelength and the returned value is in units of absolute length rather than percentage.</p> <p>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.</p>	Ref Fld	Wave	Yes

NAME	Description	Int1	Int2	Hxy, Pxy
DIST	Distortion in waves contributed by surface "Surf". This is the third order distortion calculated from the Seidel coefficients (see "Seidel Coefficients" on page 164), and is not valid for non-paraxial systems. If the surface value is zero, the distortion is given in percent instead (see "Field Curvature/Distortion" on page 156 for a detailed definition). If the absolute flag 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.	Surf	Wave	(see left)
DIVB	Divides the value of any prior operand by any factor.	Oper#	-	factor
DIVI	Division of first by second operand. The two arguments are the row numbers of the operands to divide.	Op #1	Op #2	-
DLTN	Delta N. Computes the difference between the maximum and minimum index of refraction on axis for a gradient index surface. 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".	Surf	Wave	-
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 surface "Surf" to be greater than the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.	Surf	-	-
DMLT	Diameter less than. This boundary operand constrains the diameter of surface "Surf" to be less than the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.	Surf	-	-
DMVA	Diameter value. This operand constrains the diameter of surface "Surf" to be equal to the specified target value. The diameter is twice the semi-diameter value displayed on the main spreadsheet.	Surf	-	-
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.	-	Wave	Yes
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.	-	Wave	Yes
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.	-	Wave	Yes
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.	-	Wave	Yes
EFFL	Effective focal length in lens units. This is paraxial, and may not be accurate for non-paraxial systems.	-	Wave	-

NAME	Description	Int1	Int2	Hxy, Pxy
EFLX	Effective focal length in the local x plane of the specified range of surfaces at the primary wavelength.	First surf	Last surf	-
EFLY	Effective focal length in the local y plane of the specified range of surfaces at the primary wavelength.	First surf	Last surf	-
EFNO	Effective F/#. This operand computes the Effective F/# of any field point. For a discussion of Effective F/# see "Effective F/#" on page 140. Samp is the grid size, where 10 would yield a 10 x 10 grid of rays. Wave is the integer wavelength number, Hx is used to define the field number, and Hy specifies if polarization is used (0 for no, 1 for yes). See also RELI.	Samp	Wave	See Text
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.	-	-	-
EQUA	Equal operand. This operand constrains all operands within the specified range of operands 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.	First	Last	-
ETGT	Edge thickness greater than. This boundary operand constrains the edge thickness of surface "Surf" to be greater than the specified target value. The edge thickness is calculated at the semi-diameter radius 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.	Surf	Code	-
ETLT	Edge thickness less than. This boundary operand constrains the edge thickness of surface "Surf" to be less than the specified target value. The edge thickness is always calculated at the semi-diameter radius 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.	Surf	Code	-
ETVA	Edge thickness value. Constrains the edge thickness of surface "Surf" to be equal to the specified target value. The edge thickness is always calculated at the semi-diameter radius 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.	Surf	Code	-
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 any wavelength. The value is generalized to return reasonable results even for non-rotationally symmetric systems; see "Field Curvature/Distortion" on page 156.	-	Wave	Hx,Hy
FCGT	Generalized field curvature, tangential; see FCGS.	-	Wave	Hx,Hy

NAME	Description	Int1	Int2	Hxy, Pxy
FCUR	Field curvature in waves contributed by the specified surface. If the surface value 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.	Surf	Wave	-
FICL	Fiber coupling efficiency for single mode fibers. The sampling defines the grid size used for the integration; with 1 being 32 x 32, 2 being 64 x 64, etc. The wavelength must be monochromatic, and the wavelength number specified in the Int2 column. The Hx value is the integer field position number. If Hy is zero, then the object source fiber is considered; if Hy is non zero, the object source fiber is ignored. Px and Py are used to define the source and receiver fiber NA's respectively. The calculated value is the total coupled energy efficiency, relative to unity. See "Fiber Coupling Efficiency" on page 174 for details. See also FICP.	Sam- pling	Wave	(see left)
FICP	Fiber coupling as computed using the Physical Optics Propagation (POP) algorithm, using whatever the current default settings are for the POP feature.  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" on page 530. See also FICL.	Surf	-	(see left)
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 data option "difference" MUST be selected to return valid data. The operand FOUC will return the RMS difference between the computed and reference shadowgrams. Using this operand, the optical system wavefront aberrations may be optimized to produce the reference shadowgram.	-	-	-
GBPD	Gaussian beam (paraxial) divergence in the optical space following the specified surface. If Hx is non-zero, then the computation is for the x-direction beam, otherwise, it is for the y-direction. The Hy value is used to define the input beam waist, and Px is used to define the distance from Surface 1 to the waist location. See the Gaussian beam feature for details.	Surf	Wave	(see left)
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 for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)
GBPR	Gaussian beam (paraxial) radius of curvature in the optical space following the specified surface. See GBPD for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)
GBPS	Gaussian beam (paraxial) size in the optical space following the specified surface. See GBPD for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)
GBPW	Gaussian beam (paraxial) waist in the optical space following the specified surface. See GBPD for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)



NAME	Description	Int1	Int2	Hxy, Pxy
GBSD	Gaussian beam (skew) divergence in the optical space following the specified surface. The In# is the surface number to start the propagation. The Out# is the surface at which to compute the Skew Gaussian Beam data. Hx is the wavelength number, Hy is the field number, Px is the starting surface to waist distance in lens units, and Py is the waist size in lens units. If Py is positive, then the computation is for the y-direction beam, otherwise, it is for the x-direction. The input beam is aligned along the chief ray of the specified field. For details on the Skew Gaussian Beam feature, see "Skew Gaussian Beam" on page 189.	In#	Out#	(see left)
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 for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)
GBSR	Gaussian beam (skew) radius in the optical space following the specified surface. See GBSD for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)
GBSS	Gaussian beam (skew) size in the optical space following the specified surface. See GBSD for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)
GBSW	Gaussian beam (skew) waist in the optical space following the specified surface. See GBSD for usage of the Hx, Hy, Px, and Py values.	Surf	Wave	(see left)
GCOS	Glass cost. This operand returns the relative cost factor as listed in the glass catalog for the glass on the specified surface.	Surf	-	-
GENC	Geometric Encircled Energy (distance). This operand computes the distance in micrometers to the specified fraction of geometric encircled, ensquared, x only, or y only (enslitted) energy. Int1 specifies the pupil sampling, where 1 yields 32 x 32, 2 yields 64 x 64 etc. Int2 is the integer wavelength number; 0 for polychromatic. Hx specifies the field number. Hy is the fraction of energy desired, and must be between zero and 1, exclusive. Px is the type: 1 for encircled, 2 for x only, 3 for y only, and 4 for ensquared. Py is the reference point: 0 for chief ray, 1 for centroid, 2 for vertex, 3 for middle of the spot. GENC always scales by the diffraction limited data. See also GENF, DENC, DENF, and XENC.	Sam- pling	Wave	(see left)
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. The options and settings are identical to GENC, except Hy, which here is used as the distance at which the fraction of energy is desired. See also GENC, DENC, DENF, and XENC.	Sam- pling	Wave	(see left)
GLCA	Global x-direction orientation vector component of surface "Surf".	Surf	-	-
GLCB	Global y-direction orientation vector component of surface "Surf".	Surf	-	-
GLCC	Global z-direction orientation vector component of surface "Surf".	Surf	-	-

NAME	Description	Int1	Int2	Hxy, Pxy
GLCR	Global Coordinate Rotation Matrix component at surface "Surf". The 3 x 3 R matrix has 9 components. If the Data# is 1, GLCR returns R[1][1], Data# 2 returns R[1][2], etc... through Data# 9 returning R[3][3].	Surf	Data#	-
GLCX	Global vertex x-coordinate of surface "Surf".	Surf	-	-
GLCY	Global vertex y-coordinate of surface "Surf".	Surf	-	-
GLCZ	Global vertex z-coordinate of surface "Surf".	Surf	-	-
GMTA	Geometric MTF average of sagittal and tangential response. The Samp value must be an integer (1, 2, ...) where 1 yields 32 x 32 sampling, 2 yields 64 x 64 sampling, etc. Wave can be a valid wavelength number, or 0 for polychromatic. The Field value must be a valid field number (1, 2, ...). Freq is the spatial frequency in cycles per mm. The parameter !Scl is a flag; if zero, then the diffraction limit will be used to scale the results (recommended) otherwise, no scaling is done. See "Using MTF operands" on page 421.	(see left).	+	+
GMTS	Geometric MTF sagittal response. See GMTA for details.	(see left).	+	+
GMTT	Geometric MTF tangential response. See GMTA for details.	(see left).	+	+
GOTO	Skips all operands between the GOTO operand line and the target operand number. Execution of the merit function will start again at the target line.	Op#	-	-
GPIM	<p>Ghost pupil image. GPIM controls the location of ghost pupils (and optionally ghost images) relative to the image plane. Double-bounce ghosts form images of the pupil, and if these images are formed near the focal plane 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.</p> <p>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 plane 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 int1 and int2 parameters are set to specific surface numbers, that specific ghost path is computed, if either or both of the int values are -1, then all possible surface combinations are considered. For example, if Int1 is 12 and int2 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.</p> <p>This same operand also can be used for detecting and controlling image ghosts (which are distinct from pupil ghosts) by changing the "mode" flag in the Hx column from 0 to 1, or to control ghost pupil magnification, by setting the mode to 2.</p> <p>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.</p>	First surface	Second surface	(see left)

NAME	Description	Int1	Int2	Hxy, Pxy
GRMN	Gradient index minimum index. This boundary operand sets the minimum allowable index of refraction value for the gradient index surface "Surf" at the wavelength number "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.	Surf	Wave	-
GRMX	Gradient index maximum index. This boundary operand sets the maximum allowable index of refraction value for the gradient index surface "Surf" at the wavelength number "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.	Surf	Wave	-
GTCE	Glass TCE. This operand returns the Thermal Coefficient of Expansion Alpha1 as listed in the glass catalog for the glass on the specified surface.	Surf	-	-
HHCN	Test for the hyperhemisphere condition. ZEMAX traces the specified ray to the specified surface, 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.	Surf	Wave	Yes
IMAE	<p>Image analysis efficiency. This operand returns the fractional efficiency 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 the Int1 field, "Field", which may be selected using the Int2 field, and "Field Size", which may be selected using the hx field.</p> <p>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 efficiency (normalized to unity) as computed by the image analysis feature.</p> <p>If Surf is 0, the surface specified by the saved settings will be used. If Surf is greater than zero, then the efficiency will be computed at the specified surface.</p> <p>If Field is 0, the field number specified by the saved settings will be used. If Field is greater than zero, then the efficiency will be computed at the specified field.</p> <p>If FldSz is 0, the Field Size specified by the saved settings will be used. If FldSz is greater than zero, then the efficiency will be computed using the specified Field Size.</p> <p>See the discussion "Optimizing with the IMAE operand" on page 428.</p>	Surf	Field	(see left)

NAME	Description	Int1	Int2	Hxy, Pxy
IMSF	<p>Image Surface. This operand dynamically changes which surface is interpreted as the "image surface" for all subsequent operands. The primary usage for this operand is to optimize image quality at intermediate surfaces. Select surface 0 to restore the image surface back to the original surface.</p> <p>If the refocus flag is 0, the specified surface becomes the new image surface and is not modified in any way. If refocus is set to 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.</p> <p>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" on page 109.</p>	Surface	Refocus	-
INDX	Index of refraction. Returns the current index at any surface and defined wavelength.	Surf	Wave	
InGT	Index "n" greater than. This boundary operand constrains the index of refraction at wavelength number "Wave" of gradient index surface "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 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.	Surf	Wave	-
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".	Surf	Wave	-
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".	Surf	Wave	-
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. This is the y-distance between the paraxial chief ray intercepts of the two extreme wavelengths defined. Not valid for non-paraxial systems.	Wave1	Wave2	-
LINV	Lagrange (or optical) invariant of system in lens units. The paraxial marginal and chief ray data are used to compute this value.	-	Wave	-
LOGE	Log base e of an operand. The argument 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.	Op #	-	-

NAME	Description	Int1	Int2	Hxy, Pxy
LOGT	Log base 10 of an operand. The argument 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.	Op #	-	-
LONA	Longitudinal aberration in lens units. This is the distance from the current to the focal plane defined by the wavelength and pupil zone. The distance is measured along the Z axis. If the pupil zone is zero, paraxial rays are used to determine the paraxial focal plane locations. If the pupil zone is greater than 0.0 and less than or equal to 1.0, real marginal rays are used to determine the focal plane locations. See AXCL. Not valid for non-paraxial systems.	Wave	-	Pupil Zone
LPTD	This boundary operand constrains the slope of the axial gradient index profile from changing signs within a gradient index component. See the section "Using gradient index operands".	Surf	-	-
MAXX	Returns the largest value within the indicated range of operands. See MINN.	First	Last	-
MCOG	Multi-configuration operand greater than. This is used to constrain values in the multi-configuration editor.	MC operand #	Config #	-
MCOL	Multi-configuration operand less than. This is used to constrain values in the multi-configuration editor.	MC operand #	Config #	-
MCOV	Multi-configuration operand value. This is used to directly target or compute values in the multi-configuration editor.	MC operand #	Config #	-
MINN	Returns the smallest value within the indicated range of operands. See MAXX.	First	Last	-
MNAB	Minimum Abbe number. This boundary operand constrains the Abbe number of surfaces between "First Surf" and "Last Surf" to be greater than the specified target value. See also MXAB. This operand only considers surfaces using model or substitute status catalog glasses.	First surf	Last surf	-
MNCA	Minimum center thickness air. This boundary operand constrains each of the center thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MNCG	Minimum center thickness glass. This boundary operand constrains each of the thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MNCT	Minimum center thickness. This boundary operand constrains each of the center thicknesses of surfaces from "First surf" to "Last surf" to be greater than the specified target value. See also MNCG and MNCA. This operand controls multiple surfaces simultaneously.	First surf	Last surf	-
MNCV	Minimum curvature. This boundary operand constrains each of the curvatures of surfaces from "First surf" to "Last surf" to be greater than the specified target value. See also MXCV. This operand controls multiple surfaces simultaneously.	First surf	Last surf	-

NAME	Description	Int1	Int2	Hxy, Pxy
MNDT	Minimum diameter to thickness ratio. Controls the minimum allowable value on the ratio of surface diameter to center thickness. Only surfaces with non-unity index of refraction are considered. See also MXDT. This operand controls multiple surfaces simultaneously.	First surf	Last surf	-
MNEA	Minimum edge thickness air. This boundary operand constrains each of the edge thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MNEG	Minimum edge thickness glass. This boundary operand constrains each of the edge thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MNET	Minimum edge thickness. This boundary operand constrains each of the edge thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MNIN	Minimum index at d-light. This boundary operand constrains the Nd value of surfaces between "First Surf" and "Last Surf" to be greater than the specified target value. See also MXIN. This operand only considers surfaces using model or substitute status catalog glasses.	First surf	Last surf	-
MNPD	Minimum $\Delta P_{g,F}$ . This boundary operand constrains the deviation of the partial dispersion of surfaces between "First Surf" and "Last Surf" to be greater than the specified target value. See also MXPDP. This operand only considers surfaces using model or substitute status catalog glasses.	First surf	Last surf	-
MNSD	Minimum semi-diameter. Constrains the semi-diameter to be larger than the specified target over the surface range.	First surf	Last surf	-
MSWA	Modulation square-wave transfer function, average of sagittal and tangential. See MTFT for details.	Sam- pling	Wave	(see left)
MSWS	Modulation square-wave transfer function, sagittal. See MTFT for details.	Sam- pling	Wave	(see left)
MSWT	Modulation square-wave transfer function, tangential. See MTFT for details.	Sam- pling	Wave	(see left)
MTFA	Modulation transfer function, average of sagittal and tangential. See MTFT for details.	Sam- pling	Wave	(see left)
MTFS	Modulation transfer function, sagittal. See MTFT for details.	Sam- pling	Wave	(see left)

NAME	Description	Int1	Int2	Hxy, Pxy
MTFT	Modulation transfer function, tangential. This computes the diffraction MTF. Samp must be an integer (1, 2, ...) where 1 yields 32 x 32 sampling, 2 yields 64 x 64 sampling, etc. Wave can be a valid wavelength number, or 0 for polychromatic. Field must be a valid field number (1, 2, ...). Freq is the spatial frequency in cycles per mm. 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 the discussion "Using MTF operands" in this chapter for details.	Sampling	Wave	(see left)
MXAB	Maximum Abbe number. This boundary operand constrains the Abbe number of surfaces between "First Surf" and "Last Surf" to be less than the specified target value. See also MNAB. This operand only considers surfaces using model or substitute status catalog glasses.	First surf	Last surf	-
MXCA	Maximum center thickness air. This boundary operand constrains each of the thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MXCG	Maximum center thickness glass. This boundary operand constrains each of the center thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MXCT	Maximum center thickness. This boundary operand constrains each of the center thicknesses of surfaces from "First surf" to "Last surf" to be less than the specified target value. See also MXCG and MXCA. This operand controls multiple surfaces simultaneously.	First surf	Last surf	-
MXCV	Maximum curvature. This boundary operand constrains each of the curvatures of surfaces from "First surf" to "Last surf" to be less than the specified target value. See also MNCV. This operand controls multiple surfaces simultaneously.	First surf	Last surf	-
MXDT	Maximum diameter to thickness ratio. Controls the maximum allowable value on the ratio of surface diameter to center thickness. Only surfaces with non-unity index of refraction are considered. See also MNDT. This operand controls multiple surfaces simultaneously.	First surf	Last surf	-
MXEA	Maximum edge thickness air. This boundary operand constrains each of the edge thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MXEG	Maximum edge thickness glass. This boundary operand constrains each of the edge thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-

NAME	Description	Int1	Int2	Hxy, Pxy
MXET	Maximum edge thickness. This boundary operand constrains each of the edge thicknesses of surfaces from "First surf" to "Last surf" 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.	First surf	Last surf	-
MXIN	Maximum index at d-light. This boundary operand constrains the Nd value of surfaces between "First Surf" and "Last Surf" to be less than the specified target value. See also MNIN. This operand only considers surfaces using model or substitute status catalog glasses.	First surf	Last surf	-
MXPD	Maximum $\Delta P_{g,F}$ . This boundary operand constrains the deviation of the partial dispersion of surfaces between "First Surf" and "Last Surf" to be less than the specified target value. See also MNPD. This operand only considers surfaces using model or substitute status catalog glasses.	First surf	Last surf	-
MXSD	Maximum semi-diameter. Constrains the semi-diameter to be less than the specified target over the surface range.	First surf	Last surf	-
NORD	Normal distance to the next surface. This operand computes the surface normal vector at any coordinate, then returns the distance to the next surface measured along the normal vector.	Surf	-	X, Y
NORX	Normal vector x component. This operand returns the x component of the surface normal vector at any coordinate.	Surf	-	X, Y
NORY	Normal vector y component. This operand returns the y component of the surface normal vector at any coordinate.	Surf	-	X, Y
NORZ	Normal vector z component. This operand returns the z component of the surface normal vector at any coordinate.	Surf	-	X, Y
NPGT	Non-sequential parameter greater than. The Hx value is used to define the parameter number.	Surf	Object	(see left).
NPLT	Non-sequential parameter less than. The Hx value is used to define the parameter number.	Surf	Object	(see left).
NPVA	Non-sequential parameter value. The Hx value is used to define the parameter number.	Surf	Object	(see left).
NPXG	Non-sequential object position x greater than. 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.	Surf	Object	(see left)
NPXL	Non-sequential object position x less than. See NPXG.	Surf	Object	(see left)
NPXV	Non-sequential object position x value. See NPXG.	Surf	Object	(see left)
NPYG	Non-sequential object position y greater than. See NPXG.	Surf	Object	(see left)
NPYL	Non-sequential object position y less than. See NPXG.	Surf	Object	(see left)
NPYV	Non-sequential object position y value. See NPXG.	Surf	Object	(see left)
NPZG	Non-sequential object position z greater than. See NPXG.	Surf	Object	(see left)



NAME	Description	Int1	Int2	Hxy, Pxy
NPZL	Non-sequential object position z less than. See NPXG.	Surf	Object	(see left)
NPZV	Non-sequential object position z value. See NPXG.	Surf	Object	(see left)
NSDC	Non-sequential coherent intensity data. Detector refers to the object number of the desired detector. If the object number corresponds to a detector object, then the data from the specified pixel is returned. Data is 0 for real, 1 for imaginary data, 2 for the amplitude sum, and 3 for the incoherent intensity. See "Optimizing with sources and detectors in non-sequential mode" on page 427 for complete details.	Surf	Detector	(see left)
NSDD	Non-sequential incoherent intensity data. Detector refers to the object number of the desired detector. If the object number is zero, then all detectors are cleared. The detector object may be a detector rectangle or a faceted detector object. If the object number corresponds to a detector object, then the data from the specified pixel is returned. If the pixel number is zero, then the sum of the flux or average flux/area for all pixels for that detector object is returned. If the pixel number is -1, then the maximum flux or flux/area is returned. If the pixel number is -2, then the minimum flux or flux/area is returned. If the pixel number is -3, the number of rays striking the detector is returned. If the pixel number is -4, the standard deviation (RMS from the mean) of all the pixel data is returned. Data is 0 for flux, 1 for flux/area, and 2 for flux/solid angle pixel. Only data values 0 and 1 (for flux and flux/area) are supported for faceted detectors. See "Optimizing with sources and detectors in non-sequential mode" on page 427 for complete details.	Surf	Detector	(see left)
NSRA	Non-sequential single ray trace. Source 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 Upol 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 optimized. The Dat# refers to the data for the specified segment. Use Dat# 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 Dat# 10-15 for path-to, intensity, phase of, phase at, index, and starting phase, respectively. Use Dat# 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 Dat# 31-39 for the coordinate data defined by Dat# 1-9 converted to coordinates relative to the global coordinate reference surface. For more information on these data items see "Ray database files" on page 366.	Surf	Source	(see left)

NAME	Description	Int1	Int2	Hxy, Pxy
NSTR	Non-sequential trace. Source refers to the object number of the desired source. If source 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 Upol 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" on page 427 for complete details.	Surf	Source	(see left)
NTXG	Non-sequential object tilt about x greater than.	Surf	Object	-
NTXL	Non-sequential object tilt about x less than.	Surf	Object	-
NTXV	Non-sequential object tilt about x value.	Surf	Object	-
NTYG	Non-sequential object tilt about y greater than.	Surf	Object	-
NTYL	Non-sequential object tilt about y less than.	Surf	Object	-
NTYV	Non-sequential object tilt about y value.	Surf	Object	-
NTZG	Non-sequential object tilt about z greater than.	Surf	Object	-
NTZL	Non-sequential object tilt about z less than.	Surf	Object	-
NTZV	Non-sequential object tilt about z value.	Surf	Object	-
OBSN	Object space numerical aperture. This is only useful for finite conjugate systems, and is calculated on axis at the primary wavelength.	-	-	-
OFF	This operand indicates an unused entry in the operand list. OFF operands are automatically converted to BLNK operands upon evaluation of the merit function. OFF 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.	-	Wave	Yes
OPDM	Optical path difference with respect to the mean OPD; this operand computes the OPD referenced to the mean OPD of all rays in the pupil. OPDM has the same restrictions that TRAC does; see TRAC for a detailed discussion.	-	Wave	Yes
OPDX	Optical path difference with respect to the shifted and tilted reference sphere that minimizes the RMS wavefront error; which ZEMAX calls the centroid reference. OPDX has the same restrictions that TRAC does; see TRAC for a detailed discussion.	-	Wave	Yes
OPGT	Operand greater than. This is used to make any operand an inequality constraint.	Op #	-	-
OPLT	Operand less than. This is used to make any operand an inequality constraint.	Op #	-	-
OPVA	Operand value. This operand constrains the value of a prior operand to be equal to the target value.	Op #	-	-
OPTH	Optical path length. This is the distance, in lens units, the specified ray travels to the surface "Surf". 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.	Surf	Wave	Yes

NAME	Description	Int1	Int2	Hxy, Pxy
OSCD	Offense against the sine condition (OSC). There are two definitions for OSC supported. The first definition is as described in Welford, Aberrations of Optical Systems (see "REFERENCES ON LENS DESIGN" on page 31). This definition is used if the zone parameter 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 the zone parameter is not zero. The two methods will give very similar results for systems with modest F/#'s and aberrations when the zone parameter is 1.0 for the alternate definition. This operand has no meaning if the system is not axially symmetric.	-	Wave	Zone
OSUM	Sums the values of all operands between the two specified operands. See SUMM.	First	Last	-
PANA	Paraxial ray x-direction surface normal at the ray-surface intercept. This is the x component of the surface normal vector at the intersection point of the specified paraxial ray and the surface "Surf", in the local coordinate system.	Surf	Wave	Yes
PANB	Paraxial ray y-direction surface normal at the ray-surface intercept. This is the y component of the surface normal vector at the intersection point of the specified paraxial ray and the surface "Surf", in the local coordinate system.	Surf	Wave	Yes
PANC	Paraxial ray z-direction surface normal at the ray-surface intercept. This is the z component of the surface normal vector at the intersection point of the specified paraxial ray and the surface "Surf", in the local coordinate system.	Surf	Wave	Yes
PARA	Paraxial ray x-direction cosine of the ray after refraction from the surface "Surf".	Surf	Wave	Yes
PARB	Paraxial ray y-direction cosine of the ray after refraction from the surface "Surf".	Surf	Wave	Yes
PARC	Paraxial ray z-direction cosine of the ray after refraction from the specified surface.	Surf	Wave	Yes
PARR	Paraxial ray radial coordinate in lens units at the specified surface. This is the radial distance from the local axis to the intersection of the surface "Surf" and the specified paraxial ray, in the local coordinate system.	Surf	Wave	Yes
PARX	Paraxial ray x-coordinate in lens units at the surface "Surf".	Surf	Wave	Yes
PARY	Paraxial ray y-coordinate in lens units at the surface "Surf".	Surf	Wave	Yes
PARZ	Paraxial ray z-coordinate in lens units at the surface "Surf".	Surf	Wave	Yes
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 "Surf".	Surf	Wave	Yes
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 "Surf".	Surf	Wave	Yes
PETC	Petzval curvature in inverse lens units. Not valid for non-paraxial systems.	-	Wave	-
PETZ	Petzval radius of curvature in lens units. Not valid for non-paraxial systems.	-	Wave	-

NAME	Description	Int1	Int2	Hxy, Pxy
PIMH	Paraxial image height at the paraxial image plane for the specified wavelength. Not valid for non-paraxial systems.	-	Wave	-
PLEN	Path length. This operand computes the total optical path length (including index of refraction and phase surfaces) between surfaces 1 and 2 for the specified ray, which is always traced at the primary wavelength. PLEN is essentially the difference between two OPTH operands. See OPTH.	Surf 1	Surf 2	Yes
PMAG	Paraxial magnification. This is the ratio of the paraxial chief ray height on the paraxial image plane to the object height. Only useful for finite conjugate systems. Note the paraxial image plane is used even if the system is not at paraxial focus.	-	Wave	-
PMGT	Parameter greater than. This boundary operand constrains the value of a parameter 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.	Surf	Param	-
PMLT	Parameter less than. This boundary operand constrains the value of a parameter 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.	Surf	Param	-
PMVA	Parameter value. This boundary operand constrains the value of a parameter 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.	Surf	Param	-
PnGT	This operand is obsolete, use PMGT instead.	Surf	-	-
PnLT	This operand is obsolete, use PMLT instead.	Surf	-	-
PnVA	This operand is obsolete, use PMVA instead.	Surf	-	-

NAME	Description	Int1	Int2	Hxy, Pxy
POPD	<p>Physical Optics Propagation Data. For important details see "PHYSICAL OPTICS PROPAGATION" on page 515.</p> <p>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.</p> <p>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.</p> <p>The data number determines what data the POP feature will compute and return as follows:</p> <p>0: The total fiber coupling.  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.  Undefined data numbers will return 0.  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 Extra1 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 Extra1 value. The circle is centered on the beam centroid (50), chief ray (51), or surface vertex (52).  The Extra1 and Extra2 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 surface #, wave #, field #, Extra1, and Extra2 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.</p>	Surf	Wave	(see left)
POWR	The surface power (in inverse lens units) of the surface "Surf". This operand only works for standard surfaces.	Surf	Wave	-
PRIM	Primary wavelength. This is used to change the primary wavelength number during merit function evaluation. This operand does not use the target or weight columns.	-	Wav #	-
PROB	Multiplies the value of any prior operand by any factor.	Oper#	-	factor

NAME	Description	Int1	Int2	Hxy, Pxy
PROD	Product of two operands. The two arguments are the row numbers of the operands to multiply. See PROB.	Op #1	Op #2	-
QSUM	Quadratic sum. This operand squares and then adds all operands between the first and last operand (inclusive), then takes the square root of the sum. See also SUMM, OSUM, EQUA.	First	Last	-
RAED	Real ray angle of exitance. This is the angle in degrees between the surface normal and the ray after refraction or reflection. See also RAID.	Surf	Wave	Yes
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 that surface. Does not return correct results if the glass prior to the surface is a gradient index medium. See also RAIN.	Surf	Wave	Yes
RAGA	Global ray x-direction cosine. This is the direction cosine of the ray in the global coordinate system. The origin of the global coordinate system is at the global reference surface.	Surf	Wave	Yes
RAGB	Global ray y-direction cosine. See RAGA.	Surf	Wave	Yes
RAGC	Global ray z-direction cosine. See RAGA.	Surf	Wave	Yes
RAGX	Global ray x-coordinate. This is the coordinate in lens units in the global coordinate system. The origin of the global coordinate system is at the global reference surface.	Surf	Wave	Yes
RAGY	Global ray y-coordinate. See RAGX.	Surf	Wave	Yes
RAGZ	Global ray z-coordinate. See RAGX.	Surf	Wave	Yes
RAID	Real ray angle of incidence. This is the angle in degrees between the surface normal and the incident ray. Note the angle of incidence is always positive. See also RAED.	Surf	Wave	Yes
RAIN	Real ray angle of incidence. This is the cosine of the angle between the surface normal and the ray before refraction at that surface. Does not return correct results if the surface prior to surf is a gradient index surface. See also RAEN.	Surf	Wave	Yes
RANG	Ray angle in radians with respect to z axis. The angle is measured with respect to the local Z axis.	Surf	Wave	Yes
REAA	Real ray x-direction cosine of the ray after refraction from the surface "Surf".	Surf	Wave	Yes
REAB	Real ray y-direction cosine of the ray after refraction from the surface "Surf".	Surf	Wave	Yes
REAC	Real ray z-direction cosine of the ray after refraction from the surface "Surf".	Surf	Wave	Yes
REAR	Real ray radial coordinate in lens units at the surface "Surf".	Surf	Wave	Yes
REAX	Real ray x-coordinate in lens units at the surface "Surf".	Surf	Wave	Yes
REAY	Real ray y-coordinate in lens units at the surface "Surf".	Surf	Wave	Yes
REAZ	Real ray z-coordinate in lens units at the surface "Surf".	Surf	Wave	Yes

NAME	Description	Int1	Int2	Hxy, Pxy
RELI	Relative illumination. This operand computes the relative illumination of any field point 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. Samp is the grid size, where 10 would yield a 10 x 10 grid of rays. Wave is the integer wavelength number, Hx is used to define the field number, and Hy specifies if polarization is used (0 for no, 1 for yes). See also EFNO.	Samp	Wave	See Text
RENA	Real ray x-direction surface normal at the ray-surface intercept.	Surf	Wave	Yes
RENB	Real ray y-direction surface normal at the ray-surface intercept.	Surf	Wave	Yes
RENC	Real ray z-direction surface normal at the ray-surface intercept.	Surf	Wave	Yes
RETX	Real ray x-direction ray tangent (slope).	Surf	Wave	Yes
RETY	Real ray y-direction ray tangent (slope).	Surf	Wave	Yes
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 glass selection" on page 424 for a complete discussion. The constraint is active over the surface range specified.	First Surf	Last Surf	See Text
RSCE	RMS spot radius (ray aberrations) with respect to the geometric image centroid, measured in lens units. This operand is similar to RSCH, except the reference point is the image centroid instead of the chief ray. See RSCH for details.	Rings	Wave	Hx, Hy
RSCH	RMS spot radius (ray aberrations) with respect to chief ray. This operand uses a Gaussian quadrature method to estimate the RMS spot radius at a specified field coordinate and wavelength. The number returned is in lens units. The method used is only accurate for systems with circular pupils. The Int1 column is used to specify the number of rings of rays traced (use no more than required to converge the result). Only Hx and Hy are used to define the field point, Px and Py are not used. If the "wave" value is zero, then a wavelength weighted polychromatic calculation is performed.	Rings	Wave	Hx, Hy
RSRE	RMS spot radius (ray aberrations) with respect to the geometric image centroid, measured in lens units. This operand is similar to RSCE, except a rectangular grid of rays is used instead of the Gaussian quadrature method. This operand always considers vignetting. A grid value of 1 will trace 4 rays, 2 will trace a 2 x 2 grid per quadrant (16 rays), 3 will trace 3 x 3 rays per quadrant (36 rays), and so on. Symmetry is considered.	Grid	Wave	Hx, Hy
RSRH	Similar to RSRE, except the reference point is the chief ray.	Grid	Wave	Hx, Hy
RWCE	RMS wavefront error with respect to the diffraction centroid. This operand is useful for minimizing the wavefront variance, which is proportional the Strehl ratio and the area under the MTF curve. The units are waves. See RWCH. See RSCH for details.	Rings	Wave	Hx, Hy
RWCH	RMS wavefront error with respect to the chief ray. The units are waves. This RMS really refers to the standard deviation of the wavefront, since the mean OPD is subtracted out. See RWCE. See RSCH for details.	Rings	Wave	Hx, Hy

NAME	Description	Int1	Int2	Hxy, Pxy
RWRE	Like RSRE, except for wavefront error rather than spot radius.	Grid	Wave	Hx, Hy
RWRH	Like RSRH, except for wavefront error rather than spot radius.	Grid	Wave	Hx, Hy
SAGX	The sag in lens units of the surface "Surf" in the XZ plane at the semi-diameter distance. See also SSAG.	Surf	-	-
SAGY	The sag in lens units of the surface "Surf" in the YZ plane at the semi-diameter distance. See also SSAG.	Surf	-	-
SFNO	Sagittal working F/#, computed at any defined field and wavelength. See TFNO.	Field	Wave	-
SINE	Sine of the value of the specified operand number. If flag is 0, then the units are radians, otherwise, degrees.	Op #	Flag	-
SKIS	Skip if symmetric. If the lens is rotationally symmetric, then computation of the merit function continues at the specified operand number.	Op #		
SKIN	Skip if not symmetric. See SKIS.	Op #		
SPCH	Spherochromatism in lens units. This is the difference between the real marginal axial color and the paraxial axial color. The distance is measured along the Z axis. The pupil zone defines the zone for which the real marginal axial color is computed. Not valid for non-paraxial systems.	Wave1	Wave2	Pupil Zone
SPHA	Spherical aberration in waves contributed by the specified surface. If the surface value is zero, the sum for the entire system is used.	Surf	Wave	-
SQRT	Square root of operand. The argument is the row number of the operand value to take the square root of.	Op #	-	-
SSAG	The sag in lens units of the surface "Surf" in the XZ plane at any (x, y) coordinate. The x and y values are defined in the Hx and Hy columns. See also SAGX, SAGY.	Surf	-	x, y
SUMM	Sum of two operands. The two arguments are the row numbers of the operands to add. See OSUM.	Op #1	Op #2	-
SVIG	Sets the vignetting factors. When included, updates the vignetting factors for the current configuration.	-	-	-
TANG	Tangent of the value of the specified operand number. If flag is 0, then the units are radians, otherwise, degrees.	Op #	Flag	-
TCGT	Thermal Coefficient of expansion greater than. This boundary operand constrains the TCE of surface "Surf" to be greater than the specified target value.	Surf	-	-
TCLT	Thermal Coefficient of expansion less than. This boundary operand constrains the TCE of surface "Surf" to be less than the specified target value.	Surf	-	-
TCVA	Thermal Coefficient of expansion value. This boundary operand constrains the TCE of surface "Surf" to be equal to the specified target value.	Surf	-	-
TFNO	Tangential working F/#, computed at any defined field and wavelength. See SFNO.	Field	Wave	-



NAME	Description	Int1	Int2	Hxy, Pxy
TGTH	Sum of glass thicknesses from first to last specified surface. Note that the sum is inclusive, it is not the thickness between the two surfaces. See TTHI.	First surf	Last surf	-
TMAS	Total mass. Computes the mass of the glass lenses within the specified range of surfaces. 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" on page 220 for a discussion of how element masses and volumes are computed.	First surf	Last surf	-
TOLR	Tolerance data. The "Data" value is 0 for RSS estimated change in performance, 1 for nominal performance, and 2 for estimated performance (nominal plus estimated change). For details, see "Optimizing tolerance sensitivity" on page 426.	Data	-	-
TOTR	Total track (length) of lens in lens units.	-	-	-
TRAC	Transverse aberration radial direction measured in the image plane with respect to the centroid. 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 Default Merit Function tool, and is not recommended for use directly by the user.	-	Wave	Yes
TRAD	The x component of the TRAR only. TRAD has the same restrictions that TRAC does; see TRAC for a detailed discussion.		Wave	Yes
TRAE	The y component of the TRAR only. TRAE has the same restrictions that TRAC does; see TRAC for a detailed discussion.		Wave	Yes
TRAI	Transverse aberration radius measured at the specified surface with respect to the chief ray. Similar to TRAR, except a surface other than the image surface may be specified.	Surf	Wave	Yes
TRAR	Transverse aberration radial direction measured in the image plane with respect to the chief ray. See ANAR.	-	Wave	Yes
TRAX	Transverse aberration x direction measured in the image plane with respect to the chief ray.	-	Wave	Yes
TRAY	Transverse aberration y direction measured in the image plane with respect to the chief ray.	-	Wave	Yes
TRCX	Transverse aberration x direction measured in the image plane with respect to the centroid. See TRAC. This operand should only be entered into the Merit Function Editor by the Default Merit Function tool, and is not recommended for use directly by the user.	-	Wave	Yes
TRCY	Transverse aberration y direction measured in the image plane with respect to the centroid. See TRAC. This operand should only be entered into the Merit Function Editor by the Default Merit Function tool, and is not recommended for use directly by the user.	-	Wave	Yes

NAME	Description	Int1	Int2	Hxy, Pxy
TTGT	Total thickness greater than. This boundary operand constrains the total thickness (including both front and back surface sags) of surface "Surf" to be greater than the specified target value. The thickness is calculated at the semi-diameter radius 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.	Surf	Code	-
TTHI	Sum of thicknesses from first to last specified surface. Note that the sum is inclusive, it is not the thickness between the two surfaces. See TGTH.	First surf	Last surf	-
TTLT	Total thickness less than. See TTGT.	Surf	Code	-
TTVA	Total thickness value. See TTGT.	Surf	Code	-
UDOP	User defined operand. Used for optimizing numerical results computed in externally compiled programs. See "User defined operands" on page 429. See also ZPLM.	Macro #	Data#	Yes
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" on page 389.	-	-	-
VOLU	Volume of element(s) in cubic cm. Computes the volume of the lenses and air spaces for the specified range of surfaces. 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" on page 220 for a discussion of how element masses and volumes are computed.	First surf	Last surf	-
WFNO	Working F/#. See "Working F/#" on page 55, and ISFN, SFNO, and TFNO.	-	-	-
WLEN	Wavelength. This operand returns the specified wavelength in micrometers.	-	Wave#	-
XDGT	Extra data value greater than. The Int2 number must be between 1 and 200 to indicate which of the extra data values is selected.	Surf	Number	-
XDLT	Extra data value less. The Int2 number must be between 1 and 200 to indicate which of the extra data values is selected.	Surf	Number	-
XDVA	Extra data value. The Int2 number must be between 1 and 200 to indicate which of the extra data values is selected.	Surf	Number	-
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 setting that is overwritten is the type, which is 1 for encircled, 2 for x only, 3 for y only, 4 for ensquared, 5 for x distribution, and 6 for y distribution.  Hx is the fraction of energy desired, and must be between zero and 1, exclusive. Hx is ignored for type 5 or 6; for these types the returned value is the full width half max independent of Hx. See also XENF, DENC, DENF, GENC, and GENF.	Type	-	(see left)

NAME	Description	Int1	Int2	Hxy, Pxy
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 options and settings are identical to XENC, except Hx, which here is used as the distance at which the fraction of energy is desired. See also XENC, GENC, GENF, DENC, and DENF.	Sampling	Wave	(see left)
XNEA	Minimum edge thickness for air surfaces. 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. See MNEA.	First surf	Last surf	-
XNEG	Minimum edge thickness for glass surfaces. 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. See MNEG.	First surf	Last surf	-
XNET	Minimum edge thickness. 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. See MNET.	First surf	Last surf	-
XXEA	Maximum edge thickness for air surfaces. 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. See MXEA.	First surf	Last surf	-
XXEG	Maximum edge thickness for glass surfaces. 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. See MXEG.	First surf	Last surf	-
XXET	Maximum edge thickness. 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. See MXET.	First surf	Last surf	-
YNIP	YNI-paraxial. This number is the product of the paraxial marginal ray height times the index times the angle of incidence. This quantity is proportional to the narcissus contribution of the specified surface. See Applied Optics, Vol. <b>21</b> , 18, p3393.	Surf	Wave	-

NAME	Description	Int1	Int2	Hxy, Pxy
ZERN	<p>Zernike Fringe coefficient. The Int1, Int2, Hx, Hy, and Px data values are used to specify the Zernike term number (1-37 for fringe, 1- 231 for standard), the wavelength number, the sampling density (1 = 32 x 32, 2 = 64 x 64, etc.), the field position, and the Zernike type (0 for fringe, 1 for standard), respectively.</p> <p>The term number, if negative or zero, may also be used to return other data from the Zernike fitting as follows:</p> <ul style="list-style-type: none"> <li>-8: Peak to Valley OPD (to centroid)</li> <li>-7: Peak to Valley OPD (to chief)</li> <li>-6: RMS to zero reference (unused by ZEMAX)</li> <li>-5: RMS to chief ray</li> <li>-4: RMS to centroid</li> <li>-3: Variance</li> <li>-2: Strehl Ratio</li> <li>-1: RMS fit error</li> <li>0: Maximum single point fit error</li> </ul> <p>Note that if you use multiple ZERN operands which only differ in the term number, they should be placed on adjacent lines in the editor so ZEMAX only does the fitting once; otherwise, the computation is slower.</p>	Term	Wave	(see left)
ZPLM	Used for optimizing numerical results computed in ZPL macros. See "User defined operands" on page 429. See also UDOP.	Macro #	Data#	Yes
ZTHI	This operand controls the variation in the total thickness of a range of surfaces 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.	First surf	Last surf	-

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" on page 423.

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 by selecting Tools, Update. 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.

## **Using MTF operands**

The MTF operands such as MTFT, MTFs, and MTFA provide a capability to directly optimize the diffraction MTF. This is a powerful capability, however, using the MTF operands requires some care on the part of the user.

For systems not close to the diffraction limit, geometric equivalent MTF operands are provided: GMTT, GMTS, and GMTA. These should be used in place of the diffraction operands for systems with greater than about 2-5 waves of aberration.

The MTF operands calculate the complete diffraction or geometric MTF exactly like the plots available on the Analysis, Diffraction menu option. Therefore, any optical system for which the MTF plot produces invalid data (due to excessive OPD in the pupil, see the chapter "Analysis" for details) will also produce meaningless data during optimization. For example, it is unrealistic to optimize a lens for MTF starting from plane parallel plates, because of course the MTF generally cannot be accurately computed for such a system. Also, MTF optimization is considerably slower than RMS spot radius or RMS wavefront error optimization, typically 5 to 50 times slower. Note that if you use both a MTFT and a MTFS operand 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 for accurate computation of the MTF, then the MTF operands return zero rather than a meaningless number.

The slow execution speed may be noticeable when updating the Merit Function Editor display, and entering and exiting the optimization dialog box. At these times the merit function is updated. If several MTF operands have been entered, ZEMAX may take several minutes on a slow computer just to refresh the screen.

A good approach is to interactively design your system using RMS wavefront error. Generally speaking, systems with low RMS wavefront error will have reasonable MTF performance. After the design is very nearly in the final form, then try switching over to MTF optimization for a "touch-up". It is also a good idea to completely erase the default merit function before entering any MTF operands, except of course for any boundary constraints needed (such as MNCT).

The MTF operands use the operand data columns such as Int1, Int2, and Hx differently than most operands. The Int1 column determines the sampling density to be used. A value of 1 specifies a 32 x 32 grid should be used, 2 specifies a 64 x 64 grid, and so on. Use the smallest grid size for which the data are computed accurately; again see the "Analysis" chapter for details.

The Int2 column selects the wavelength, like most operands. However, a value of zero may be entered to specify a polychromatic calculation. In this case, the wavelength-weighted polychromatic MTF is computed. Obviously, this is slower than a monochromatic MTF calculation.

The Hx column is used to specify the field position, and it must be an integer between 1 and the number of defined fields.

The Hy column is the spatial frequency in cycles per millimeter, independent of the lens system units. Any value may be entered; the operand will return 0 if it is past the incoherent cut-off frequency. The value does not need to be an integral multiple of some fraction; the exact value of the MTF is computed using a cubic spline fit to the neighboring data points, just like the MTF data plots.

The target and weight columns are used just like any other operand. Of course, if the target is set to 1.0, and the frequency is non-zero, then the value of the operand can never reach the target.

## **Performing an optimization**

To begin optimization, choose Tools, Optimization from the main menu bar. The optimization control dialog box will appear with the following options.

### OPTIMIZATION OPTIONS

Item	Description
Automatic	Executes until ZEMAX determines the system is no longer improving significantly.
1 Cycle	Executes a single optimization cycle.
5 Cycles	Executes 5 optimization cycles.
10 Cycles	Executes 10 optimization cycles.
50 Cycles	Executes 50 optimization cycles.
Inf. Cycles	Executes optimization cycles in an infinite, continuous loop until "Terminate" is pressed.
Terminate	Terminates a running optimization, and returns control back to the dialog box.
Exit	Closes the optimization dialog box.

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.
# CPU's	Selects the number of CPU's 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.

Selecting automatic will cause the optimizer to run until no progress is being made. The other options will run the specified number of cycles. 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 Terminate to end the optimization run.

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.

## **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 adds 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 glass selection**

Optimization of glasses 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 or by using glass substitution. Glass substitution is usually far superior, but is only supported in the EE edition of ZEMAX.

### **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. This is the "model" glass method. Model glasses are described in detail in the chapter "Using Glass Catalogs". 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. This method is used by the conventional optimizer described in this chapter.



To optimize glasses then requires several steps. First, change the glass of the appropriate surface to a "Model" glass using the glass solve dialog box in the Lens Data Editor. For information on model glasses, see "Using model glasses" on page 485. When you change the glass from "Fixed" to "Model", ZEMAX will make a suitable guess for the index, Abbe number, and partial dispersion; you only need to change the values if you wish. The three values can be made variable by clicking on the "Vary" box next to each entry.

The Ctrl-Z shortcut on the glass column will also work; it makes the index, Abbe, and partial dispersion variable automatically. The model glass data values can now be optimized using the optimization feature in the usual way.

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 Nd higher than about 1.9. The Vd value also is limited to the range of roughly 20 to 80. Therefore, it is essential to limit the Nd and Vd values to reasonable ranges during optimization. The partial dispersion deviation 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 (these are specified on the general data window), 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 preceding tables. 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 Ctrl-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" on page 195. 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 Summary" feature (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.

### 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 using the global optimization techniques described in the next chapter, "Global Optimization". The global optimization method uses only actual glass catalog materials, and in this sense is the superior method. See "Using glass substitution" on page 437 for a discussion on this method for optimizing glass selection.

### Optimizing zoom and multi-configuration lenses

Optimizing zoom lenses is virtually identical to optimizing conventional single-configuration lenses. See the Chapter "Multi-Configurations" for details.

### Optimizing extra data



*This discussion is only relevant for users of ZEMAX-EE.*

---

Certain surface types supported by ZEMAX-EE, 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 ASCII files, and may be used as variables for optimization. For details on the editor, see the chapter "Editors menu".

To make an extra data value a variable, open up the Extra Data Editor. When the extra data editor appears, move the cursor to the row and column with the value you want to optimize, and press Ctrl-Z (the same command used to set variables on the main screen). The variable will now be optimized when the optimization is run.

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 Int1 column on the spreadsheet display indicates the surface number to which the operand applies, and the Int2 column is used to specify which of the extra data values is to be used.

### 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" on page 441.

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 criteria 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. The only criteria that is not allowed is "Merit Function" since that would create an infinite loop. If a script tolerance criteria 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 criteria 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 criteria, as this will create an infinite loop.

The faster the tolerance sensitivity runs, the faster TOLR and thus the optimization will execute.

### **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 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 NSDD 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 NSDD or NSDC 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. If pixel is -1, the maximum flux or flux/area for all pixels is returned. If pixel is -2, the

minimum flux or flux/area for all pixels is returned. If pixel is -3, the number of rays striking the detector is returned. If pixel is -4, the RMS deviation from the mean for all pixels is returned. The units of the returned data is determined by the system units, see "Analysis Units" on page 87. 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 exists for coherent data using NSDC.

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 rectangular detectors, the pixels are numbered starting at pixel number 1 in the (-X, -Y) corner. The pixel numbers increase across the rows, which run in the +X direction. If there are Nx pixels in the X direction, then pixel number Nx will be located at the end of the first row, which is the (+X, -Y) corner. The next pixel number, Nx+1, will be the first pixel of the second row, located at (-X, -Y+dY), where dY is the distance between rows (i.e. the distance between pixels in the y-direction). Pixels continue to be numbered in this manner across successive rows. The (-X, +Y) corner will be pixel number ((Nx\*(Ny-1))+1). The (+X, +Y) corner will be the last pixel, pixel number (Nx\*Ny).

For other detectors, such as polygon objects used as detectors, 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.

### **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 Chapter "Surface Types" under "User Defined Surface" 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 "Geometric Image Analysis" on page 142.

### **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. Int1 is used to define the surface number, and Int2 is used to define the wavelength number. DLTN is defined as:

$$DLTN = n_{max} - n_{min} \cdot$$

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. Only the Int1 value 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.

## **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" on page 423.

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. However, it is limited by the capabilities of the ZPL macro language, and ZPL macros are interpreted, which means slow execution for complex calculations. 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 since external programs are compiled, they run significantly faster. The speed difference can be dramatic, and generally more complex calculations will benefit more from being externally compiled. Indeed, externally defined operand programs may be very complex, tracing millions of rays or doing other lengthy calculations before returning control to ZEMAX. Note this interface can be used to optimize lenses based upon data computed by other analysis programs.

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" on page 535) 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 Int1 and Int2 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 Int1 value for a ZPLM operand is set to 17, for example, then the macro number is 17, and 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 default directory for ZPL macros; see "Directories" on page 59.

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 plane (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 Int1 = 15 and Int2 = 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 Hx, Hy, Px, and Py data fields. 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 Hx, Hy, Px, and Py values can only be used if the Int2 value is zero, since only in this case is the macro evaluated.

Lastly, it is very important that no changes be made to the lens data during the macro execution. These changes may interfere with the subsequent evaluation of other operands. ZEMAX does not restore the lens being evaluated to the state it was in prior to the evaluation of the ZPLM specified macro. Also, 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. If lens data is changed during the operation of the macro, ZEMAX has no way of knowing what data was altered, and cannot restore the lens to its original unaltered state.

This could be avoided by allowing the ZPL macro to execute only on a copy of the lens being optimized, rather than the actual lens, however this is not currently supported. The reason is there may be times where a macro

needs to alter the lens data prior to evaluation of subsequent operands. In this case, 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.

### 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" on page 607. 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 Int1 and Int2 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 the Int1 value for a UDOP operand is set to 17, for example, then the client program number is 17, and the client program 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 \UDO directory off the main ZEMAX directory.

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 hx hy px py
```

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 51 values are placed in a long formatted string. The string is sent back to the server, and ultimately to the optimizer within ZEMAX, with the SetUDOData item. The syntax is "SetUDOData, buffercode, data0, data1, data2, data3, ...,data50". Any data items omitted are assumed to be zero. 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 SetUDOData, even if it only includes the buffercode with the remaining fields blank. 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 for dataflag is 0, 1, and 2 */
/* DO NOT use SetUDOData if the total line length will exceed 255 characters, */
/* use SetUDOIItem instead. See the chapter on ZEMAX Extensions for details. */
sprintf(szBuffer, "SetUDOData,%i,%.7f,%.7f,%.7f", buffer_code, a, b, c);
PostRequestMessage(szBuffer, szBuffer);
}
```

Note that multiple data values may be returned with a single call to the client program. This is where the data field number comes into play. 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 client program, the client DDE item "SetUDOData" is used to return a long string of 51 numbers (or less; blank or omitted values are assumed to be zero). These values, which must be numeric, are stored in the lens buffer for later use by UDOP. In practice, the 255 character limitation of DDE item names limits SetUDOData to passing back far fewer than 51 items; but the DDE item "SetUDOIItem" works around this limitation by passing back data one at a time.



***There is a limitation of 255 characters in any DDE item name; and so SetUDOData is limited to passing back a small number of values. To work around this limitation, use SetUDOIItem, described in the chapter "ZEMAX Extensions".***

There are 51 different data fields, so that a single client program call can be used to optimize up to 51 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 Hx, Hy, Px, and Py data 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. 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 "SetUDOData,buffercode,x,y,z". 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 Hx, Hy, Px, and Py values can only be used if the data field value is zero, since only in this case is the client program evaluated.

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 UDOfx.EXE where xx is a two digit integer. The sample UDO returns 6 values: Hx, Hy, Px, Py, and two dummy constants, in data positions 0 through 5, respectively.

## **Suggestions for use**

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 in the next chapter.



## **Introduction**



*The material presented in this chapter critically depends upon the user having read and understood the material presented in the previous chapter on "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  $1\text{E}+12$  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  $8\text{E}+07$  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  $1\text{E}+32$  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.

## **Capabilities of ZEMAX**

There are two separate global optimization algorithms in ZEMAX, each with a different purpose. The first algorithm you are likely to use is called "Global Search", and it is used to find new design forms given only the merit function and a starting design. Global search uses a combination of genetic algorithms, multistart, conventional damped least squares, and some expert system heuristics to search for new design forms. The Global Search 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). The Hammer 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 Search algorithm. The Hammer 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 Search and Hammer algorithms require extensive computational effort to be effective. These algorithms are not intended to be used interactively! (that is what DLS 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 Search (or Hammer, depending upon your requirements) work overnight. In the morning, you should have useful results to look at.

## **The Global Search algorithm**

Before you begin the Global Search, 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 "OPTIMIZATION" on page 385 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 Search. ZEMAX uses the starting focal length as a scaling parameter, so the initial design should have at least the approximately correct focal length!

From the main screen, select Tools, Global Search. There are four buttons on the dialog box labeled: Start, Stop, Resume, and Exit. There is also an option to select the number of "best" files to save. Select Start. 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 search 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.

To terminate the search, select Stop. 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.

The Resume button 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 the search anew 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.

Global search will rarely find the global optimum by itself. The reason is that the Global Search 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 algorithm**

After reviewing the design forms generated by Global Search, 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 following controls are on the Hammer dialog box:

**Hammer:** 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.

**Auto DLS:** Calls the conventional (local) damped least squares (DLS) optimizer and runs it in Automatic mode. Sometimes it is useful to optimize a lens before calling Hammer if the lens has not already been optimized.

**Stop:** Terminates the Hammer search.

**Exit:** Closes the dialog box.

**Auto Update:** If checked, all open windows will be updated every time a lower merit function is found.

**#CPU:** The number of CPU's to use in implementing the Hammer search.

Hammer optimization only requires a ZMX file with variables and a merit function to get started. 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 Hammer 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.

The Hammer algorithm can also be used effectively on partially optimized designs that were not generated with Global Search. Feel free to use the Hammer on any design!

## **Optimizing 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**

ZEMAX automates this procedure 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 Search) 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 Search 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 on the System, General dialog box.

### **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 on the System, General dialog box. 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 "Glass Substitution Template" on page 195 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 the Chapter "Using Glass Catalogs" for details. 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 use**

There are several techniques for maximizing the performance of the global optimization algorithms (both Global Search and Hammer):

1) If possible, place the stop on the first surface. If your entrance pupil is imbedded 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 plane. 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 Search is in focus, just as the curvature solve ensures the correct focal length. These two solves together can improve the performance of the Global Search 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) You may want to dim or turn off your monitor, or use a screen saver to protect the phosphor from burn-in if you plan on running the search for many days.

9) The Hammer algorithm may be terminated and then restarted without significant loss of information. As the algorithm proceeds, it automatically saves any improved designs.

10) If a Global Search is terminated, the search may be resumed at a later time by choosing resume instead of start.

11) While the optimizer is running, it will dominate the computer's resources. Other applications will still run, but they will be less responsive. Under Windows NT/2000, the resources given to ZEMAX may be reduced by opening the Task Manager; then choosing the ZEMAX.EXE Process; right mouse click; then choose Set Priority -> LOW.

12) 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 Search 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.





## **Introduction**

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 on the main menu bar.

Tolerances may be evaluated by several different criteria, 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 criteria is determined for each tolerance individually. Optionally, the criteria for each field and configuration individually may be computed.

**Inverse Sensitivity:** For a given permissible change in criteria, the limit for each tolerance is individually computed. Inverse sensitivity may be computed by placing a limit on the change in the criteria from nominal, or by a limit on the criteria directly. The criteria 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 criteria. 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 this chapter is a good place to start. Tolerances are defined and modified on the Tolerance Data Editor, available on the Editors menu of the main menu bar.
- 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 plane tilt, may be defined. There is no limit to the number of compensators that may be defined.
- 4) Select an appropriate criteria, such as RMS spot radius, wavefront error, MTF, or boresight error. More complex criteria 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 criteria 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

Name	Int1	Int2	Int3	Description	Page
Surface Tolerances					
TRAD	Surf #	-	-	Tolerance on radius in lens units	443
TCUR	Surf #	-	-	Tolerance on curvature in inverse lens units	444
TFRN	Surf #	-	-	Tolerance on radius in fringes	444
TTHI	Surf #	Adjust #	-	Tolerance on thickness or position in lens units, with optional adjuster.	444
TCON	Surf #	-	-	Tolerance on conic constant (dimensionless)	444
TSDX	Surf #	-	-	Tolerance on Standard surface x-decenter in lens units	445
TSDY	Surf #	-	-	Tolerance on Standard surface y-decenter in lens units	445
TSTX	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in x in degrees	445
TSTY	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in y in degrees	445
TIRX	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in x in lens units	445
TIRY	Surf #	-	-	Tolerance on Standard surface tilt (TIR) in y in lens units	445
TIRR	Surf #	-	-	Tolerance on Standard surface irregularity	445
TEXI	Surf #	# of Terms	-	Tolerance on surface irregularity using Fringe Zernikes	446
TEZI	Surf #	# of Terms	-	Tolerance on surface irregularity using Standard Zernikes	446

Name	Int1	Int2	Int3	Description	Page
TPAR	Surf #	Parameter #	-	Tolerance on parameter value of surface	448
TEDV	Surf #	Extra Data #	-	Tolerance on extra data value of surface	448
TIND	Surf #	-	-	Tolerance on index of refraction at d light (see notes).	448
TABB	Surf #	-	-	Tolerance on Abbe Vd number (see notes)	448
TCMU	Surf #	Layer #	-	Tolerance on coating multiplier	448
Element Tolerances					
TEDX	First Surf	Last Surf	-	Tolerance on element x- decenter in lens units	448
TEDY	First Surf	Last Surf	-	Tolerance on element y- decenter in lens units	448
TETX	First Surf	Last Surf	-	Tolerance on element tilt about x in degrees	449
TETY	First Surf	Last Surf	-	Tolerance on element tilt about y in degrees	449
TETZ	First Surf	Last Surf	-	Tolerance on element tilt about z in degrees	449
User Defined Tolerances					
TUDX	Surf #	-	-	Tolerance on user defined decenter x	449
TUDY	Surf #	-	-	Tolerance on user defined decenter y	449
TUTX	Surf #	-	-	Tolerance on user defined tilt x	449
TUTY	Surf #	-	-	Tolerance on user defined tilt y	449
TUTZ	Surf #	-	-	Tolerance on user defined tilt z	449
Non-Sequential Component Tolerances					
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	449
TNPA	Surf #	Object #	Parameter	Tolerance on NSC object parameter	449
Multi-Configuration Value Tolerances					
TMCO	Row #	Config#	-	Tolerance on multi-configuration editor value	450

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 tolerated.

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.

### *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. See the discussion "Tolerancing with the Irregular surface type" on page 450.

### *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.

See the description of the related TIR operands TIRX and TIRY.

The analysis of TSTX and TSTY uses the Irregular surface type. See the discussion "Tolerancing with the Irregular surface type" on page 450.

### *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. See the discussion "Tolerancing with the Irregular surface type" on page 450.

### *TIRR: Tolerance on surface irregularity*



***For a more detailed treatment of irregularity, see 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

$$\Delta z = \frac{\lambda_t W}{4} (\rho^4 + \rho_y^2),$$

where  $\lambda_t$  is the test wavelength (defined by the TWAV operand),  $\rho$  is the normalized radial coordinate, and  $\rho_y$  is the normalized radial coordinate in the y direction. 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:

$$\Delta OP = \Delta z \left( \frac{n_2}{n_1} - 1 \right).$$

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.

See the discussion "Tolerancing with the Irregular surface type" on page 450 for more information.

### **TEXI: Tolerance on surface irregularity using the Fringe Zernike model**



**See also the discussion for TEZI on page 447. 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. The Int1 value indicates the number of the surface, and the Int2 value defines the number of Fringe Zernike terms used to simulate the irregularity on the surface. Analysis of irregularity on surface types other than Standard, Even Aspheric, or Zernike Fringe Sag is not directly supported.

TEXI uses the Zernike Fringe Sag surface (see "Zernike Fringe Sag" on page 283) 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:

$$c = \frac{\lambda f}{2\sqrt{n}},$$

where  $f$  is the number of double pass fringes,  $n$  is the number of Zernike terms used in the irregularity model, and  $\lambda$  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 either the min or the max "c" value from the above equation. Note that since the surface sag deformation is the same for all 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; see the STAT command for a discussion.

The default number of Zernike terms is 37, the maximum. Generally speaking, if fewer terms are used, the irregularity will be of low frequency, with fewer "bumps" across the surface. If more 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 37 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 0, 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 on page 446. 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 or Even Aspheric surface. The Int1 value indicates the number of the surface, and the Int2 value defines the number of Standard Zernike terms used to simulate the irregularity on the surface. Analysis of irregularity on surface types other than Standard or Even Aspheric surfaces is not directly supported.

TEZI uses the Zernike Standard Sag surface (see "Zernike Standard Sag" on page 285) to model the irregularity. 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 on the Zernike Standard Sag surface. 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 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 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 zero and 1, 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; see the STAT command for a discussion.

The number of Zernike terms may be between 0 and 231, the maximum. Generally speaking, if fewer terms are used, the irregularity will be of low frequency, with fewer "bumps" across the surface. If more 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 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 TEZI operand. The normalization radius of the Zernike Fringe Sag surface is set to the semi-diameter of the surface.

TEZI always ignores Zernike term 0, the piston term, and sets this value to zero.

### **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. See "SURFACE TYPES" on page 225 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 the extra data editor in ZEMAX-EE 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. See the Chapter "Surface Types" 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.

### **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 elsewhere, see "Optimizing coatings with ZEMAX" on page 508 and "Surface coating tab" on page 74. 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.

### **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, see 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, see the SAVE command discussed in the section "Trouble shooting the tolerance results".



TEDX and TETY 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" on page 462.

### *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, see 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" on page 462.

### *TOFF: Tolerance off (can be used for comments)*

This operand is a place holder that has no affect 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. It is important that the coordinate breaks inserted for use with the TUxx tolerances have nominal values of zero for all tilts and decenters. 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: 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. See “NSC Objects” on page 293 for a detailed explanation of what parameter numbers are used by which object types.

### **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. This operand is ignored unless all configurations are tolerated at once.

### **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, see page 262.

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**

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. See the discussion on CMCO below for important information.
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, see “SURFACE TYPES” on page 225.
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.

See "SURFACE TYPES" on page 225 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 criteria, although the compensators will still be used.

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.

See "MULTI-CONFIGURATIONS" on page 471 for information on using multi-configuration operands.

See also "General comments about min and max values on compensators" on page 451.

### **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)

Int2 = 2, compensator is conic

See also "General comments about min and max values on compensators" on page 451.

### **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.

See "SURFACE TYPES" on page 225 for tables defining which parameters are used by which surface types.

See also "General comments about min and max values on compensators" on page 451.

### **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 is 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 0 or any other digit. The next time the tolerance analysis is run, ZEMAX will save the file used to compute the TEDX tolerance in a file named TSAVE000n.ZMX where n is the integer number specified in the Int1 column.

The lens file will be saved in the same directory as the current lens. ZEMAX will save the lens used to evaluate the preceding tolerance in a ZMX format file. Run the tolerance file in the usual way, and then load the filename

you specified. It will show you the prescription data, the coordinate breaks, pickups, variables (which are optimally adjusted compensators), and merit function used to determine the criteria data (usually for the maximum tolerance value). 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.

The statistics types are defined in detail in "Monte Carlo analysis" on page 460.

### **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 default tolerances**

The default tolerances may be defined by selecting Tools, Default Tolerances from the Tolerance Data Editor menu bar. The Tolerance Data Editor may be activated by selecting Editors, Tolerance Data from the main menu.

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" on page 444.

**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 TEDI 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 TETI 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 in degrees is given by

$$\theta_x = \frac{\Delta_y}{2S}, \text{ where}$$

S is the semi-diameter of the surface and  $\Delta$  is the total indicator runout, with a similar expression for  $\theta_y$ . 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 the TIRR description given earlier.

Zern Irreg: If checked, a Zernike irregularity is specified on each Standard surface type. For details, see the TEXI description given earlier.

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.

## **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" on page 449 for information on tipping about some other point.

In addition to the tolerance definitions there are two other options on the dialog box:

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.

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.

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.

## **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 pick ups 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, see "Using Tolerance Scripts" on page 463.

## **Preparing the lens for tolerancing**

Generally speaking, no changes are required to the lens before starting the tolerance procedure. There are however a few exceptions.

The tolerance algorithm works most reliably if the criteria being used consists only of image quality evaluation operands; such as the RMS spot radius, RMS wavefront, or MTF. If a user-defined merit function is used, some care must be taken in the operands selected for use in the merit function.

Merit functions that contain construction constraints that reference individual surface numbers can cause incorrect results. The problem arises because ZEMAX must insert dummy surfaces to automatically tilt and decenter surfaces and groups of surfaces. This insertion causes a renumbering of the operands in the merit function which may or may not be correct.

For example, to tilt a range of surfaces, ZEMAX must insert 3 dummy surfaces, and then copy the thickness of the previous last surface in the range to the new dummy surface which follows the range. This can cause the operands which previously constrained the thickness on one surface to no longer constrain the correct value or surface.

A similar problem arises when negative or small thickness values are not allowed by the merit function constraints; since the insertion of the dummy surfaces may violate these boundary constraints.

The recommended solution is to use only merit functions that do not include surface-specific constraints. If these operands must be used; then the surfaces targeted should be dummy surfaces in between lens elements; rather than surfaces which are part of a lens group.

Note that for tolerancing purposes it is not usually required to constrain negative edge or center thicknesses, or other mechanical packaging constraints. These unphysical conditions, which must be prevented during optimization, are not a concern after the lens is physically fabricated.

## **Performing the tolerance analysis**

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.

Mode: Four modes are supported: Sensitivity, Inverse Limit, Inverse Increment, and Skip Sensitivity.

Sensitivity mode computes the change in the criteria for each of the extreme values of the tolerances.

Inverse Limit computes the value of each tolerance that will yield a criteria 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" on page 459.

Inverse Increment computes the value of each tolerance that will yield a *change* in the criteria 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. See "Inverse sensitivity analysis" on page 459.

Skip Sensitivity will bypass the sensitivity analysis and proceed to the Monte Carlo analysis.

Limit: When using Inverse Limit mode, this control is active and is used to define the limit on the criteria for computing inverse tolerances. For example, suppose the Criteria 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" on page 459. The nominal value for the currently selected criteria may be computed by pressing the "?" 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 criteria for computing inverse tolerances. For example, suppose the Merit 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 an RMS spot radius 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" on page 459. The nominal value for the currently selected criteria may be computed by pressing the "?" button adjacent to the Increment edit window.

**# 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.

**Save MC Runs:** 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.

**Criteria:** This control is used to specify what shall be used as the criteria 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 criteria. 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" on page 389 for a discussion.

**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 criteria. 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.

**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" on page 463.

**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 always measured in line pairs per mm in the image space.

**Sampling:** Sampling is used to set how many rays are traced when computing the tolerance criteria. Higher sampling traces more rays, and gives more accurate results. However, the execution time increases. If the selected criteria 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" on page 387 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 criteria, 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.

**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.

**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. If "Paraxial Focus" is selected, only the change in paraxial back focus error is considered as a compensator; all other compensators are ignored. Using Paraxial Focus is very useful for rough tolerancing, and is significantly faster than using "Optimize All".

**# Opt 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".

**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.

**Separate Fields/Configs:** If checked, the criteria will be computed and displayed for all field positions in all configurations individually. If unchecked, the criteria 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 criteria, 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 criteria 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 criteria at every field is degraded no more than the increment value. See "Inverse sensitivity analysis" on page 459.

Script: The name of the script, if using the User Script criteria. User scripts must be ASCII files ending in the extension TSC and be in the same directory as the ZEMAX.EXE program executable.

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".

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 criteria for each tolerance.

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.

Overlay MC Graphics: If checked, each open analysis graphical window (such as a ray fan or MTF graph) will be updated and overlaid 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. Overlaid 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.

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. 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.

Show worst: See "Hide all but worst" above.

Status: This control is used by the tolerance algorithm to provide status messages during computation of the tolerance analysis.

There are also six buttons on the bottom of the dialog box:

OK: Performs the tolerance analysis using the current options.

Cancel: Exits the dialog box without performing the tolerance analysis.

Terminate: Terminates the tolerance analysis.

Save: Save the currently selected options for future use.

Load: Restore the previously saved settings.

Reset: Restore the settings to the default values.

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.

## **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; see "Tolerancing with solves" on page 468 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. See "Ray Aiming" on page 88 for details on ray aiming.

ZEMAX then uses the Criteria, 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 criteria of this lens is considered to be the "nominal". Note that the nominal criteria value generally will not be the same as the criteria 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 criteria 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 criteria 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 criteria 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.

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 criteria 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. The resulting averaged squared values are then summed for all the tolerances, and the square root of the result is taken. The average of the min and max tolerances 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 resulting RSS is the estimated change in performance.

### **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 criteria is approximately equal to the Max Criteria. For Inverse Increment mode, the adjustments are made until the resulting change in criteria is approximately equal to the Increment value.

For example, if the mode is Inverse Limit, the Criteria is RMS spot radius, the nominal criteria is 0.035, and the Max Criteria is 0.050, ZEMAX will adjust the tolerances until the criteria is 0.050. If the mode is Inverse Increment, the Criteria is RMS spot radius, the nominal criteria 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 Criteria must be greater than the nominal, except for MTF Criteria, in which case the Max Criteria 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.

If the option to "Separate Fields/Configs" is not selected, the criteria ZEMAX uses to perform the inverse analysis is the overall criteria, 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 criteria at each field in each configuration individually, and verifies that each field meets either the Max Criteria or Increment value. For Inverse Increment mode, ZEMAX computes the nominal performance at every field position and reduces the tolerances until the criteria at every field is degraded no more than the increment value.

If the starting value of the tolerance yields performance better than the Max Criteria 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 Criteria is 0.050, and the initial tolerance yields a criteria 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 one over " $n$ " times half the maximum range of the tolerances. 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  $\Delta$  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  $\Delta$  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 an ASCII 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 must be placed in the same directory as the ZEMAX program, 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 criteria 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 EE edition of ZEMAX.*

---

### **Tolerance Script overview**

Tolerance scripts are macro-like command files that define a procedure to follow for evaluating the performance 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 ASCII 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 ASCII files ending in the extension TSC, and be placed in the same directory as the ZEMAX program.

Note that when using tolerance scripts, the min and max boundary values of compensators are not enforced; see "General comments about min and max values on compensators" on page 451 for more information.

### **The Tolerance Script commands**

The tolerance script commands are defined and described below.

**!**

**Syntax:**

! A comment line!

The "!" symbol is used to define comments in the script that are ignored during execution of the script.

#### **CEDV**

**Syntax:**

CEDV surf param

CEDV defines a new extra data value compensator. The value "surf" is the surface number in the original lens file. ZEMAX automatically rennumbers 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**

**Syntax:**

CLEARCOMP

CLEARCOMP removes all current compensators. New compensators must be defined before calling OPTIMIZE.

### CLOSEFILE

#### *Syntax:*

CLOSEFILE

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

#### *Syntax:*

CMCO operand config

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.

### COMP

#### *Syntax:*

COMP surf code

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

#### *Syntax:*

CPAR surf param

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

### LOADMERIT

#### *Syntax:*

LOADMERIT filename.mf

LOADMERIT "C:\SOME\PATH\SOME MERIT FILE NAME.MF"

The MF file must reside in the same directory as ZEMAX if no path is specified. The file should be in the proper format as saved using the Tools, Save option of the Merit Function Editor.

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 "criteria" unless a subsequent LOADMERIT or OPTIMIZE command is executed.

## NOMINAL

### *Syntax:*

NOMINAL keyword

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 commands are ignored.

OPT\_ON: Turns on optimization of the compensators. Any subsequent OPTIMIZE commands are executed.

The NOMINAL command is only executed during computation of the nominal system performance, and not during the analysis of individual tolerances.

## OPENFILE

### *Syntax:*

OPENFILE "FILENAME" mode

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 ASC for an ASCII 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 command is executed. However, the text string is not included; just the numerical data the REPORT command prints to the tolerance output file. If mode is ASC, the file will be ASCII 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 criteria, 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

### *Syntax:*

OPTIMIZE n

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.

## PERTURB

### *Syntax:*

PERTURB type int1 int2 int3 stat nstd min max

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 criteria.

## **REPORT**

### ***Syntax:***

```
REPORT "text" operand  
REPORT "text" SZERNIKE term field wave sampling maxorder
```

There are two different ways to use the REPORT command:

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 (See "Zernike Standard Coefficients" on page 168). 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.

## **SAVE**

### ***Syntax:***

```
SAVE n
```

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**

### **Syntax:**

UPDATE

Updates the current lens file. 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
! finally, load the evaluation merit function, and 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 criteria which ZEMAX reports and uses.

## **Tolerancing multi-configuration (zoom) lenses**

Tolerance analysis can be performed on each configuration of a multiple configuration lens. Select the desired configuration from the "Configuration #" drop down box. Generally, it is not required to tolerance all configurations,

but that is certainly an option. When using inverse tolerances, successive tolerance analysis in each configuration 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 a case like this, 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, you should delete all solves and variables before tolerancing a lens. It may be more convenient to save the lens in a temporary file for tolerancing if you have many solves or variables to delete that you may need later. Although the tolerancing will still work if you leave solves in your lens file, the results may not be what you expect. The reason is that certain solves, such as paraxial 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. Pickup solves also can cause strange results because the tolerancing algorithm sometimes will manipulate the surface description data (such as thicknesses) when inserting and deleting coordinate breaks. The same limitation applies to position solves, which may no longer work correctly when the lens data is altered by the tolerancing algorithm.

There are however occasions where a pickup solve is really what you want. For example, if you are using a lens 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 special case, you will need to use the TUTX, and TUTY commands, and the default tolerances will not work. If you have these types of solves in the system, ignore the warning "Solves should be deleted before tolerancing".

## **Trouble shooting the tolerance results**

See the discussion in the section on the SAVE operand.

If any of the calculated tolerance data show a value of "Infinity", this means the criteria could not be evaluated for the specified tolerance. Usually the criteria 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 criteria 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" on page 426.

## **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.

Because the tolerance routine tilts and decenters lenses, the algorithm automatically turns ray aiming "on" (see "Ray Aiming" on page 88 for details about ray aiming). If the nominal criteria the tolerance algorithm reports is different from the expected value, check to see if ray aiming is "off" in the lens file. Try using ray aiming "on" and re-optimizing. Generally, if there is a big difference between criteria 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.



## **Introduction**

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.

Select Editors, Multi-Configuration from the main menu bar. The spreadsheet which appears is the multi-configuration editor (MCE). Using the menu bar on the MCE, both configurations (columns) and operands (rows) 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.

## **Summary of multi-configuration operands**

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.

SUMMARY OF MULTI-CONFIGURATION OPERANDS

Type	Numbers 1,2,3	Description
APER	Ignored	System aperture value (whatever the current aperture definition is, such as entrance pupil diameter or F/#). See also SATP.
APDX	Surface #	Surface aperture X- decenter. The surface must have a defined aperture (NOT semi-diameter).
APDT	Ignored	System apodization type. Use 0 for none, 1 for Gaussian, 2 for tangential. See also APDF.
APDF	Ignored	System apodization factor. See also APDT.
APDY	Surface #	Surface aperture Y- decenter. The surface must have a defined aperture (NOT semi-diameter).
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.

Type	Numbers 1,2,3	Description
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.
CONN	Surface #	Conic constant.
COTN	Surface #	The name of the coating, if any, to be applied to the surface.
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. The configuration weight is only used by the default merit function algorithm to set up a merit function which favors configurations via relative weighting. If the configuration weight is zero, then the configuration is ignored during construction of the default merit function.
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.
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.
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.
LTTL	Ignored	Lens title. When using LTTL, the lens title is limited to 20 characters.



Type	Numbers 1,2,3	Description
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.
NCOM	Surface, Object	Modifies the comment for non-sequential objects in the NSC Editor.
NCOT	Surface, Object, CSG#	Modifies the coating on each coating scatter group 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 flag 4 - Rays Ignore Object flag 201-212 - User defined gradient index parameters 301-312 - User defined diffraction parameters 401-412 - User defined bulk scatter parameters
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. See "Parameter data" on page 225.
PRES	Ignored	Air pressure in atmospheres. Zero means vacuum, 1 means normal air pressure. See "Environment" on page 90.
PRWV	Ignored	Primary wavelength number.
PSP1	Surface #	Parameter solve parameter 1 (the pickup surface). This operand requires 2 numerical arguments: the surface number and the parameter number.

Type	Numbers 1,2,3	Description
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 will 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.
RAAM	Ignored	Ray aiming. Use 0 for off, 1 for on.
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.
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.
TELE	Ignored	Telecentric in object space, 0 for no, 1 for yes.
TEMP	Ignored	Temperature in degrees Celsius. See "Environment" on page 90.
THIC	Surface #	Thickness of surface.
TSP1	Surface #	Thickness solve parameter 1. See "SUMMARY OF SOLVES" on page 379.
TSP2	Surface #	Thickness solve parameter 2. See "SUMMARY OF SOLVES" on page 379.
TSP3	Surface #	Thickness solve parameter 3. See "SUMMARY OF SOLVES" on page 379.
UDAF	Surface #	User defined aperture file. Surface must use either a user defined aperture or user defined obscuration aperture type. See "Surface properties aperture tab" on page 68.
WAVE	Wave #	Wavelength.
WLWT	Wave #	Wavelength weight.
XFIE	Field #	X-field value.
YFIE	Field #	Y-field value.

## **Defining the number of configurations**

The number of configurations (also called the number of zoom positions) is changed by inserting or deleting configurations using the Edit menu option.

## **Defining each configuration**

To define a multi-configuration operand, double click on the name of the operand you wish to change (the names are listed in the left most column). 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 Edit menu. Double click on the left most column in row 1. Select "THIC" from the drop-down list on the dialog box for the Operand Type. Select "5" for the surface number, then click on OK. Now in each configuration column, enter the desired thickness in that configuration.

## **Adding and removing elements**

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 making "phantom" elements in some of the configurations. The trick is to define the element (really the first surface of the element) to have a glass type which changes with the configuration. Suppose in configuration 1 the element is present and is made of BK7. In configuration 2, the element is absent. Using the GLSS operand, enter in "BK7" for configuration 1 and leave the field for configuration 2 blank. Although the surfaces which define the element are still present in the lens prescription, they have no effect because the glass has been removed (the element becomes a pair of dummy surfaces). Note that the number of surface must remain constant in all configurations.

## **Changing configurations**

To perform any analysis, proceed as usual. The program will use the current configuration for all calculations and graphics. 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 optimize across configurations, select "Default Merit Function" from the Tools menu of the merit function editor. ZEMAX will build an appropriate merit function for you. 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.



***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.***

---

## **Using solves with multi-configuration data**

Two types of solves are supported on the multi-configuration screen: a pick up from another operand and configuration, with an optional scale factor and offset, and a thermal pick up solve.

### **MCE Pick up solves**

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:  $\text{new\_value} = \text{target\_value} * \text{scale} + \text{offset}$ . 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.

### **MCE Thermal pick up solves**

ZEMAX-EE also supports a "Thermal Pick Up" solve which incorporates thermal effects; see the chapter "Thermal Analysis" for details.

## **Introduction**

There are several standard catalogs supplied with ZEMAX, and you may create your own custom catalogs. The standard catalogs may be edited to suit your requirements. This chapter describes how to add glasses to the existing or new catalogs, and how to use the catalogs in your optical designs. There is no limit to the number of glass catalogs you may create.

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 temperature defined in the glass catalog.



***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 the section "Fitting index data".



***For important comments about obsolete glasses and differences between glasses with identical names, see "Obsolete catalog data" on page 490.***

---

## **Specifying which glass catalogs to use**

This chapter describes the loading, editing, and managing of the glass catalogs. To specify that a particular catalog be used for a particular lens, use the Glass Catalog tab of the System, General dialog box. The default entry is "schott" which refers to the SCHOTT.AGF glass catalog. You may specify an alternative, or additional catalogs on this line. To specify an alternative catalog, enter the name of the catalog (with no extension). You may also specify multiple simultaneous catalogs by listing them with spaces between the names. For example, to use the "schott" and the "hoya" catalogs, enter "schott hoyo". To use the "ohara", "schott", and "infrared" catalogs, enter "ohara schott infrared". Because ZEMAX uses spaces as delimiters between glass catalog names, spaces are not allowed within the names of the glass catalog files.

All the listed catalogs must be in the \ZEMAX\GLASSCAT subdirectory of the main ZEMAX directory. 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. You can specify a different combination of catalogs for every lens you design, if you like.

## **Editing and reviewing glass catalogs**

To edit or review data in an existing glass catalog, select Tools, Glass Catalogs. 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.



***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 a great deal of 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" on page 480.
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" on page 482.
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.
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.
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.
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.

Item	Description
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 for which the index data is defined. See also "Defining temperature and pressure" on page 491.
D0, D1, D2, E0, E1, Ltk	These are the thermal coefficients used by the thermal analysis model. See the chapter "Thermal Analysis" for details.
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 Wave, Max Wave	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 (zero being best) the more durable the glass is. 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 ASCII 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 ASCII 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 directory 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” on page 485 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  $\lambda$  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” on page 482.

### **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.

## General comments of 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.

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 an ASCII file for later use, and loaded for fitting again by selecting the appropriate buttons. The ASCII 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:" on page 483) 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" on page 482) 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  $\alpha$  is the absorption coefficient and  $\tau$  is the path length through the glass. The parameter  $\alpha$  generally depends upon wavelength and has units of inverse length. See the Chapter "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.



***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.***

---

## **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 glass is to click once on the name of the glass in the Lens Data Editor, then select Tools, Glass Catalogs (or select the GLA button if displayed). The correct catalog and glass will be displayed.

## **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. You can enter a six-digit number directly in for the glass rather than the glass name if you desire.

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 .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.

## **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  $N_d$ . The Abbe number (also called the V-number) is given the symbol  $V_d$  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" on page 424 and "Optimizing glass selection" on page 437.



***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.***

---

## **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	Address	Phone	Website
ARCHER	Archer OpTx	3412 Enterprise Drive, Rowlett, TX 75088	972-463-8001	www.archeroptx.com
CORNING	Corning France Optical Division	7 bis avenue de Valvins, F- 77211 Avon Cedex, France	33-1-6469-7590	www.corning.com
CHENGDU	Chengdu Guangming	No. 720 Huan-Chen Rd., Jiading District, Shanghai China	86-28-4332218	www.xinhuglass.com
HEARAEUS	Heraeus Quarzglass GmbH & Co. KG	Quarzstrabe, D-63450 Hanau	49-0-618-1350	www.heraeus.de
HIKARI	Hikari Glass USA, Inc.	2711 Aliso Creek Road, Suite 170, Aliso Viejo, CA 92656	949-831-4908	www.hikariglass.com
HOYA	Hoya Optics, Inc.	3400 Edison Way, Fremont, CA 94538	510-252-8370	www.hoya.co.jp
LZOS		Russian optical glass fabricator	095-552-1486	www.lzos.ru/indexe.htm
OHARA	Ohara Corporation	23141 Arroyo Visa, Suite 200, Rancho Santa Margarita, CA 92688	949-858-5700	www.oharacorp.com
PILKINGTON	Pilkington Special Glass Limited	Glascoed Road, St Asaph, Clwyd, LL18, OLL, United Kingdom	44-1-7455-83301	www.pilkington.com
SCHOTT	Schott Glass Technologies	400 York Avenue, Duryea, PA 18642	570-457-7485	www.schott.com
SUMITA	Sumita	4-7-25 Harigaya, Urawa- Ku, Saitama-City, Saitama, 330-8565 Japan	81-48-832-3165	www.sumita-opt.co.jp
UMICORE	Umicore Electro-Optic Materials	Watertorenstraat 33, 2250 Olen, BELGIUM	+32 14 24 55 97	www.optics.unicore.com
ZEON	Zeon Chemicals L.P.	Furukawa Sogo Bldg., 2-6-1 Marunouchi, Chiyoda-ku, Tokyo 100-8323, Japan	33-216-2335	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. <b>23</b> , 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 ( <a href="http://www.ispoptics.com">www.ispoptics.com</a> )
KCL	ISP Optics data sheet ( <a href="http://www.ispoptics.com">www.ispoptics.com</a> )
KRS5	Handbook of Optics Vol. II
LIF	Handbook of Optics Vol. II



Material	Source
MGF2	M.J. Dodge, "Refractive Properties of Magnesium Flouride," Applied Optics, Vol. <b>23</b> , No. 11: p1980 (1985)
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

Material	Source
SEAWATER	Laikin, Lens Design
SILICA	Handbook of Optics Vol. II
TEO2	Handbook of Optics Vol. II
YPEA	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.

## **Introduction**



*This feature is only available in the EE edition of ZEMAX.*

---

Environmental factors such as ambient 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 and thickness of a lens. Third, the spacings between lens elements can change 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 ambient temperature in degrees Celsius and the ambient air pressure measured in atmospheres. These values are set on the dialog box which can be found under the Environment tab of the System, General dialog box. There is also a checkbox on this dialog box with the caption "Use Temperature, Pressure". If this checkbox is unchecked, then the temperature and pressure data are ignored. If "Use Temperature, Pressure" is checked, then the temperature and pressure values are considered during computation of the index of refraction.

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 vacuum or in a heated area; while others are not.

## **Defining wavelengths**

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 on the "Wavelength Data" dialog box; see the "Wavelengths" on page 96 for details.



*Wavelength data are always measured in micrometers referenced to "air" at the system temperature and pressure.*

---

## **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 be in air at the reference temperature of the glass.

- Compute the relative index of refraction of the glass from the catalog dispersion formulas.
- 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 specified 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 specified 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.



**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 ambient 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” on page 478. 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  $\lambda$  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:

$$\Delta n_{abs} = \frac{n^2 - 1}{2n} \left[ D_0 \Delta T + D_1 \Delta T^2 + D_2 \Delta T^3 + \frac{E_0 \Delta T + E_1 \Delta T^2}{\lambda^2 - \lambda_{tk}^2} \right],$$

where n is the relative index at the reference temperature of the glass,  $\Delta T$  is the change in temperature relative to the reference temperature of the glass ( $\Delta T$  is positive if the temperature is greater than the reference temperature of the glass), and the six other constants are provided by the glass manufacturer to describe the thermal behavior of the glass. 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" on page 495 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" on page 90. 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 the chapter 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,  $\alpha$  is the TCE, and  $\Delta T$  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 the chapter "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 "Thermal Pick Up". This pick up 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 pick up 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) 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  $23.50 \times 10^{-6}$ , 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.

### Parameter values

The thermal pick up 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 pick up 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.

### Extra data values

The thermal pick up 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 pick up 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 pick ups, 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 pick up relationships.

Checking the thermal pick up 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 "Tools" on page 78.

## 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 the chapter "Using Glass Catalogs" for the description of  $\alpha$ , 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

See "Modeling gases and liquids" on page 484 for important information on thermal modeling of non-solid materials.

## Adding thermal index variation data

The variation of index of refraction with temperature, ambient 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  $dn/dt$ .



***Extreme caution and suspicion is warranted when using only a single  $dn/dt$  value.***

---

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 pick up 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.



## **Introduction**



*This feature is only available in the EE edition of ZEMAX.*

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  $E_x$ ,  $E_y$ , and  $E_z$  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  $E_s$  and  $E_p$  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 glass 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,  $\tau_s$  and  $\tau_p$ , are complex valued. After the  $E_s$  and  $E_p$  projections are computed, they are recomposed into the  $E_x$ ,  $E_y$ , and  $E_z$  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  $\tau_s$  and  $\tau_p$  values "Field" coefficients when using this reference. Although these values are computed and displayed for comparison to thin film programs, they are not used by ZEMAX for ray tracing.

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 between the surfaces. Correctly computing the phase of the ray requires back propagating the electric field a distance equal to the physical coating thickness, and adjusting the coating phase to be measured along the ray vector rather than the normal vector. ZEMAX calls the  $\tau_s$  and  $\tau_p$  values "Ray" coefficients when using this convention, and always uses these values for ray tracing. 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. However, this conversion may be disabled, see "Convert thin film phase to ray equivalent" on page 91.

### 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 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  $\tilde{E}$  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

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

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, $\tau$

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, $\alpha$

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" on page 484.

#### Internal Transmittance, $IT$

The internal transmittance is computed using the path length  $\tau$  and internal absorption  $\alpha$ :

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

#### Propagation Phase Factors, $pc$ , $ps$

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

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

where  $\tau$  is the path length and  $\lambda$  is measured in the medium. The propagation factors are written as the cosine and sine of the angle  $\theta$ . 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" on page 501.

#### Amplitude Reflectance, $\rho$

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, $\tau$

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, $A_s$ , $A_p$

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, $P_s$ , $P_p$

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  $\pi$ :

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

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

### Phase Difference Between X and Y, $P_{xy}$

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

$$P_{xy} = P_x - P_y.$$

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

### Major and Minor semi axis of the Polarization Ellipse, $E_M$ , $E_m$

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 ( $A_p$ ) uniquely define the polarization state in the XY plane.

### Angle of XY Polarization Ellipse, $A_p$

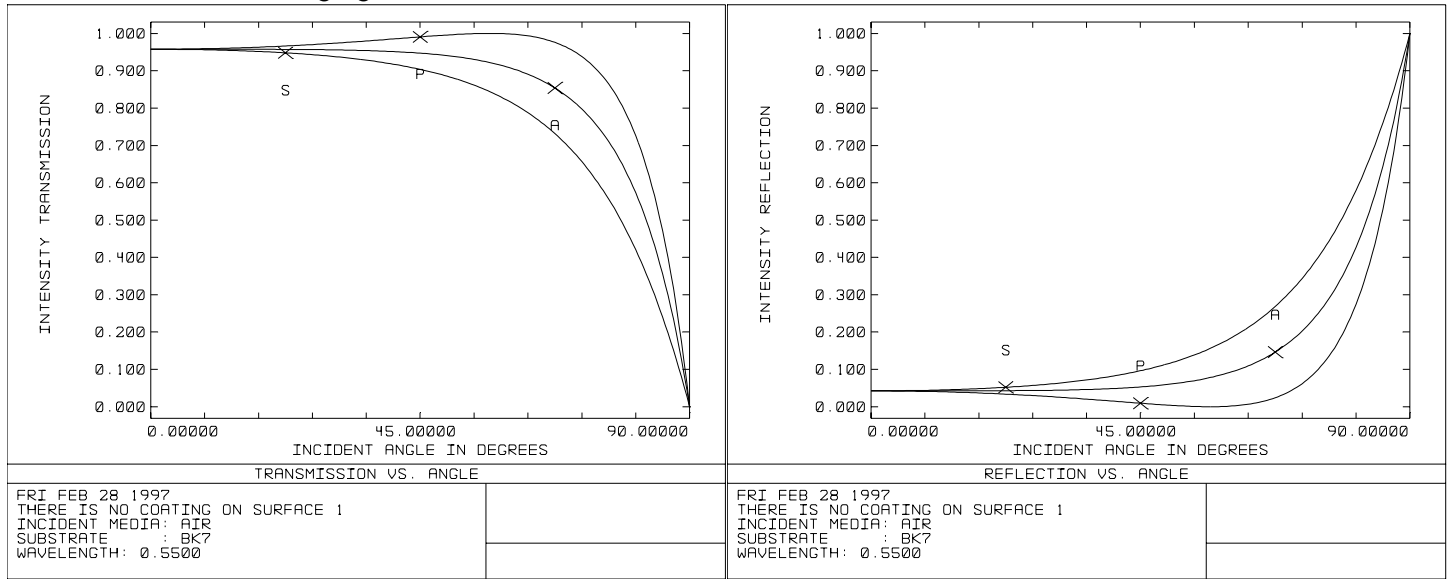
This parameter is the angle relative to the local +x axis that the major axis of the polarization ellipse makes. If the polarization ellipse is a circle, this angle is meaningless.

## **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:

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

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” on page 499.

ZEMAX uses an ASCII 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 on the System, General, Files tab dialog box. 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

! 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 spaces or 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 wavelength, 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  $f_r$ ,  $f_c$ , and  $f_p$  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  $\theta$  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:

$$f_r = \sum_{i=0}^{20} \beta_i r^i ,$$

The cosine taper factor is defined as:

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

where  $\alpha$  is defined as:

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

where  $r$  is the radial coordinate as defined above and  $R$  is the radius of curvature of the substrate.

The polynomial taper factor is defined as:

$$f_p = \sum_{i=0}^{20} \gamma_i E_i ,$$

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  $\delta_0$  cosine taper factor should be zero and the  $\delta_1$  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  $x_d$ ,  $y_d$ ,  $\theta$ ,  $\gamma_i$ ,  $\delta_i$ , and  $\beta_i$  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 spaces or 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  $\theta$ .

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,  $\beta_i$ .

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,  $\delta_i$ .

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,  $\gamma_i$ .

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 "Surface coating tab" on page 74 and "Optimizing coatings with ZEMAX" on page 508.

### The COAT data section



***The order of the coating layers is important! See the discussion in a following section.***

---

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 spaces or 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  $\lambda_0$  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 following section on coating data limits). 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. For a more general ideal coating, see the IDEAL and IDEAL2 coating definitions which follow.

### The IDEAL data section

Syntax:

```
IDEAL <name> <Transmitted intensity> <Reflected intensity>
```

Ideal coatings are defined only by the intensity transmission and reflection coefficients. Ideal coating names may be any user defined name up to 20 characters in length, with no spaces or special characters permitted. After the ideal coating name, the T and R 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. The phase of reflection is zero if the incident index is higher than the substrate index, otherwise the phase of reflection is  $\pi$ .

### 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
```

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 spaces or 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 incident index is lower than the substrate index, ZEMAX changes the sign of the reflected real and imaginary coefficients to yield an additional phase change upon reflection of  $\pi$ , 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 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 absorption is computed from  $A = 1 - R - T$  for each polarization state. If the incident index is higher than the substrate index, the phase of reflection is changed by a factor of  $\pi$ .

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. For coatings placed on surfaces going from glass to air, or on a non-sequential object, ZEMAX will automatically reverse the coating if required by computing the correct angle of incidence to use.

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.

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 incident index is higher than the substrate index, the phase of reflection is changed by a factor of  $\pi$ .

Generally much more than the brief set of data listed here is required for adequate accuracy. ZEMAX currently allows up to 20 table coatings in each coating file, each with up to 20 angle points and 100 wavelength points per table coating. If more data points are needed to model the coating, contact technical support for assistance.

#### Adding comments to the coating file

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

#### Limits on the amount of coating data

There are limits to the amount of data that may be stored in any one catalog as follows:

50 different coating names.

200 layers per coating (see the section "Defining replicated groups of coating layers" on page 506 to work around this limit if required).

50 different materials.

120 dispersion data points per material.

20 different tapers.

22 taper terms per taper.

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 ASCII 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, 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. Optimization may then be made on the usual coating performance data using the CODA operand. To optimize the layer thicknesses, 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.

Coating multipliers may then be defined on the surface coating tab described in “Surface coating tab” on page 74. This tab supports setting the multipliers to a variable status, which allows for optimization of the relative coating layer thickness.

### **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

Materials	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
Coatings	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 Listing report described in the chapter "Reports menu".

## **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" on page 512.

## **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 incident polarization**

The individual analysis settings, such as those for the polarization ray trace (see the Analysis Menu chapter) 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  $J_x$  and  $J_y$  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  $J_x$  and  $J_y$  values to have unity magnitude, and then scales the intensity as appropriate if any pupil apodization has been specified. The  $J_x$  and  $J_y$  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 ( $J_x$ ,  $J_y$ ) to the 3D electric field values ( $E_x$ ,  $E_y$ ,  $E_z$ ) is ambiguous. It is not possible to interpret the  $J_x$  and  $J_y$  values as meaning that for any ray, the  $J_x$  value should be applied in such a way to leave the  $E_y$  zero, and for the  $J_y$  to be applied so that the  $E_x$  is zero. The reason is that the  $E$  resulting for the Jones vectors ( $J_x = 1$ ,  $J_y = 0$ ) and ( $J_x = 0$ ,  $J_y = 1$ ) 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" on page 182.

## **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. To compute the E component of the electric field, the boundary condition is

$$\vec{k} \bullet \vec{E} = 0,$$

where  $\vec{E}$  is the electric field after the Jones matrix and  $\vec{k}$  is the ray vector. Since the X and Y components of  $\vec{E}$  are known, this expression can be used to compute the Z component.

This general 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	$\begin{bmatrix} (1, 0) & (0, 0) \\ (0, 0) & (0, 0) \end{bmatrix}$

Component	Matrix
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 "Analysis" chapter 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 "Analysis Menu" chapter 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  $\alpha$  is the absorption coefficient and  $\tau$  is the path length through the glass. The parameter  $\alpha$  generally depends upon wavelength and has units of inverse length. See the Chapter "Using Glass Catalogs" for details on defining transmission data. Polarization effects are not considered in all ZEMAX computations.

### **Modeling birefringent materials**

Birefringent materials are modeled using the birefringent in/out surface types, described in "Birefringent In and Birefringent Out" on page 235.

### **Modeling frustrated total internal reflection**

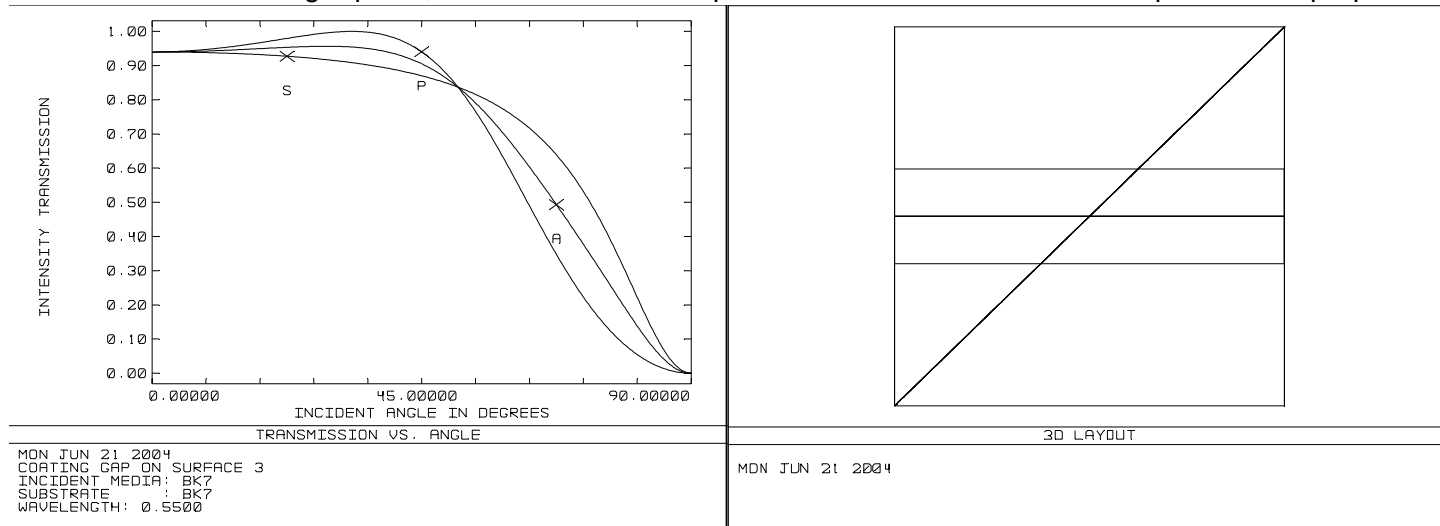
Frustrated TIR (FTIR) occurs when a ray of light traveling through glass strikes an interface at an angle exceeding the critical angle. If another dielectric medium, such as another piece of glass, is placed close, but not touching the interface, some light will transmit through the thin gap and propagate, 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 "Defining coatings in ZEMAX" on page 501.

With the GAP coating in place, ZEMAX can now compute either transmitted or reflected polarization properties.



Transmission vs. angle for a FTIR beamsplitter.

A FTIR beamsplitter in transmission. The tilted surface between the two glass blocks has a GAP coating.

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.

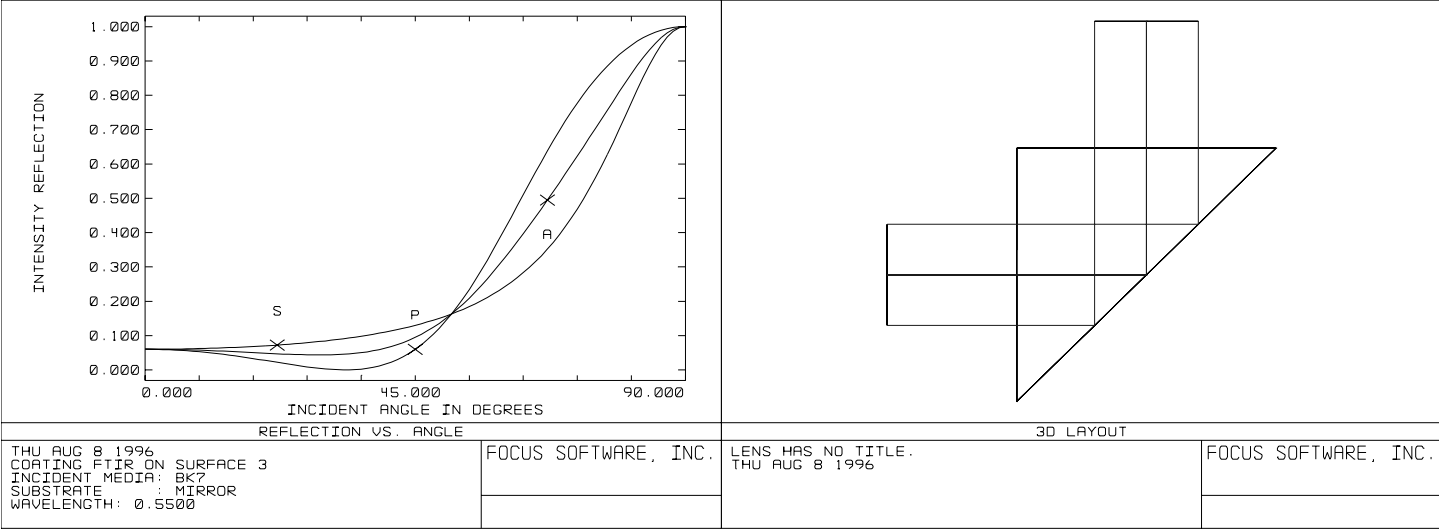
The coating in this case now needs to be modified to indicate the presence of the second dielectric block, even though no ray tracing will be conducted in the transmitted path; just the reflected path. A new coating needs to be defined in the coating file (called FTIR in this example) that looks like:

```
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.



Reflection vs. angle for a FTIR beamsplitter.

A FTIR beamsplitter in reflection. The tilted "mirror" surface has a FTIR coating.

**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.

## **Introduction**



***This feature is only available in the EE edition of ZEMAX. Physical optics propagation is complex, and the user is urged to read and understand this chapter 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 significantly 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  $\pi$ .

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  $\pi$ . 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 a 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  $\alpha$ ,  $\beta$ , and  $\gamma$ , 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:

$$FF[A] \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [A] e^{-i2\pi(x\xi + y\eta)} dx dy ,$$

$$FF^{-1}[A] \equiv \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} [A] e^{i2\pi(x\xi + y\eta)} d\xi d\eta .$$

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  $\gamma$ , 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 see “Algorithm assumptions” on page 533), 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  $\alpha$  and  $\beta$  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  $T(\Delta z)$  has unity amplitude but a complex phase. This phase varies slowly from point to point in the frequency domain representation  $G$  if  $\lambda \Delta z p^2$  is small. But if  $\lambda \Delta z p^2$  grows large, the phase variations become increasingly rapid. If the phase changes by more than about  $\pi/2$  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  $p$ ) 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  $z_2 - z_1 = \Delta z$ , then  $\Delta z$  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 see “Algorithm assumptions” on page 533).

In the Fresnel region the electric field distribution is given by

$$E(x_2, y_2, z_2) = \left[ \frac{e^{ikz}}{i\lambda\Delta z} \right] q(r_2, \Delta z) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x_1, y_1, z_1) q(r_1, \Delta z) e^{-\frac{i2\pi}{\lambda\Delta z}(x_1x_2 + y_1y_2)} dx_1 dy_1, \text{ where}$$

$$q(r, \Delta z) = e^{(i\pi r^2)/(\lambda\Delta z)}.$$

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  $q(r, \Delta z)$ , called the quadratic phase factor, indicates that the phase is referenced to a sphere of radius  $\Delta z$  (strictly speaking it is a parabola, but we have already assumed in the Fresnel development that  $r_2 \ll \Delta z$ ). 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  $q(r, \Delta z)$  is that as  $\Delta z$  gets larger,  $q(r, \Delta z)$ , varies *more slowly* in phase. This is the opposite of the  $T(\Delta z)$  operator, which varies rapidly in phase as  $\Delta z$  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  $\Delta z$  is very large, then  $q(r, \Delta z)$  may be neglected. This yields the Fraunhofer diffraction expression, which is

$$E(x_2, y_2, z_2) = \left[ \frac{e^{ikz}}{i\lambda\Delta z} \right] \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} E(x_1, y_1, z_1) e^{-\frac{i2\pi}{\lambda\Delta z}(x_1x_2 + y_1y_2)} dx_1 dy_1, \text{ or}$$

$$E(x_2, y_2, z_2) = FF(E(x_1, y_1, z_1)),$$

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  $\omega_0$ . 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  $z \rightarrow \infty$ . 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 transition point is somewhat arbitrary, however for optical design applications a value of twice the Rayleigh range has proved useful, and ZEMAX uses this threshold. For this reason, in the following discussion about propagation in and out of the Rayleigh range, the actual threshold point may be different than the Rayleigh distance. The concept however remains the same.

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 chapter.

### **Sign conventions for phase data**

It is very important to understand the phase sign conventions when constructing beams using DLL's, 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 electric 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 because 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 crossing 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  $\phi = \tan^{-1}(z/z_r)$ . 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.

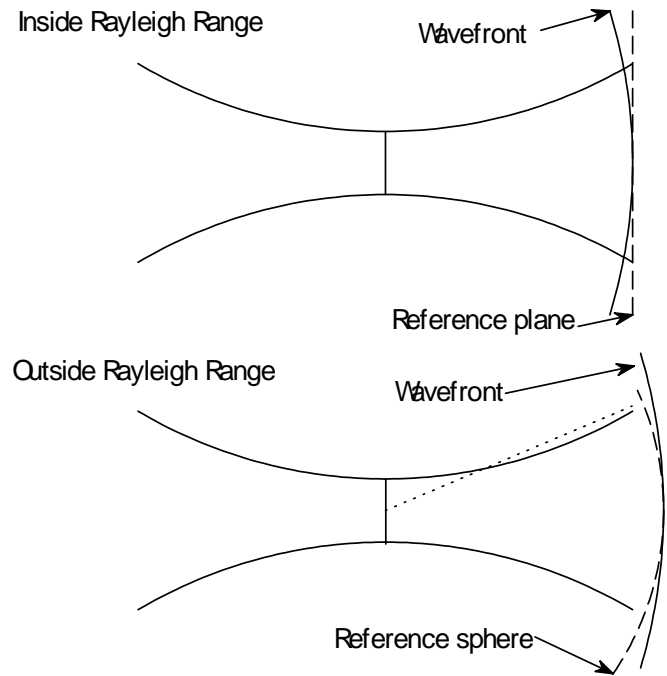
Finally, as the beam propagates further past the Rayleigh range, the phase relative to the reference sphere will tend toward a constant value of  $\pi/2$  radians (the limiting value of the Gouy shift).

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 chapter.

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:

$$E(x_2, y_2, z_2) = WTS[E(x_1, y_1, 0), z_2] \quad , \text{ where}$$

$$WTS(E, \Delta z) \equiv \left[ \frac{1}{i\lambda z_2} \right] FF^s[q(r_1, z_2)E] \quad , \quad s = \frac{z_2}{|z_2|} \quad ,$$

$$\Delta x_2 = \frac{\lambda |z_2|}{n_x \Delta x_1} \quad , \text{ and } \Delta y_2 = \frac{\lambda |z_2|}{n_y \Delta y_1} \quad ,$$

where  $n_x$  and  $n_y$  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:

$$STW(E, \Delta z) \equiv q(r_1, z_2) FF^s[i\lambda z_2 E] \quad ,$$

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:

$$II(z_1, z_2) = PTP(z_2 - z_1)$$

$$IO(z_1, z_2) = WTS(z_2 - z_0)PTP(z_0 - z_1)$$

$$OI(z_1, z_2) = PTP(z_2 - z_0)STW(z_0 - z_1)$$

$$OO(z_1, z_2) = WTS(z_2 - z_0)STW(z_0 - z_1)$$

where  $z_0$  is the pilot beam waist position,  $z_1$  is the starting beam position, and  $z_2$  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, see "Beam Definition Tab:" on page 190.

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:

$$z = \frac{c_x x^2 + c_y y^2}{1 + \sqrt{1 - c_x^2 x^2 - c_y^2 y^2}} \quad ,$$

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  $\Delta x$  and  $\Delta y$  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. See "Physical Optics Propagation" on page 190.

### **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 renormalized 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	48 bytes per grid point Some surface types require additional data and may require up to 64 bytes per grid point

For example, propagating a 2048 x 2048 unpolarized beam will require about 256 Mb. ZEMAX currently supports a minimum array size of 32 x 32 and a maximum array size of 8192 x 8192. ZEMAX requires the sampling to be an integral power of 2; for example, 32, 64, 128, 256, etc.

If the computer does not have sufficient RAM, it may "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. The Windows OS does not allow ZEMAX to allocate more than 2.0 Gb of RAM, and in many cases, the actual limit is much lower. This limit is imposed by Windows, and not ZEMAX.

## **Defining the initial beam**

The menu option Analysis, Physical Optics, Physical Optics Propagation invokes the Physical Optics Propagation 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. See "Physical Optics Propagation" on page 190.

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 defined using any of these options:

Gaussian Waist

Gaussian Angle

Gaussian Size+Angle

Top Hat

User defined table in a file

User defined DLL program

The beam may then be aligned along the chief ray for any defined field position, in the optical space preceding any surface. Propagation then proceeds from the starting surface to the ending surface. Definitions of beam waist, Rayleigh range, and divergence angle may be found in "Propagating the embedded beam" on page 188.

## **Gaussian Waist**

The Gaussian Waist beam is an optionally truncated and decentered Gaussian defined by:

$$E(x, y) = E_0 e^{-\left(\left(\frac{x-dx}{\omega_{0x}}\right)^2 + \left(\frac{y-dy}{\omega_{0y}}\right)^2\right)}, \text{ if}$$

$$\left(\frac{x-dx}{A_x}\right)^2 + \left(\frac{y-dy}{A_y}\right)^2 \leq 1, \text{ and}$$

$$E(x, y) = 0 \text{ otherwise.}$$

The dx and dy values are the decenter values, Ax and Ay are the truncating aperture values, and  $E_0$  is chosen to yield the peak irradiance in power per unit area defined in the settings box. If Ax or Ay are zero, no truncating aperture is used. A smoothing function is used near the edge of the truncating aperture to avoid 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 beam is defined at the waist, so the initial phase is zero across the full width of the beam. The beam will 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 similar to the Gaussian Waist. The distinction is that the Gaussian Angle beam is defined 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 rest of the beam definition is then identical to the Gaussian Waist.

### Gaussian Size+Angle

The Gaussian Size+Angle beam is similar to the Gaussian Waist and Angle beams. The distinction is that the Gaussian Size+Angle beam is defined by the beam size (not waist) at the starting surface 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 values are different, then an elliptical beam with a possibly toroidal phase will be generated.

### Top Hat

The top hat beam is defined as

$$E(x, y) = E_0 \text{ if } \left(\frac{x-dx}{w_x}\right)^2 + \left(\frac{y-dy}{w_y}\right)^2 \leq 1, \text{ and}$$

$$E(x, y) = 0 \text{ if } \left(\frac{x-dx}{w_x}\right)^2 + \left(\frac{y-dy}{w_y}\right)^2 \geq 1,$$

where  $E_0$  is chosen to yield the peak irradiance in power per unit area defined in the settings box.

The width of the array should be at least 6-10 times the waist size w, in each direction.

## User defined table

The table of values must be placed in either a binary or ASCII 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  $E_x^2 + E_y^2$  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\BEAMFILES subdirectory off the main ZEMAX directory.**

---

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  $(n_x/2+1, n_y/2+1)$ . ZEMAX requires the values of  $n_x$  and  $n_y$  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. 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 0.  
1 integer: The number of x samples ( $n_x$ ).  
1 integer: The number of y samples ( $n_y$ ).  
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 wavelength in lens units of the beam in the current medium.  
1 double: The waist in lens units of the pilot beam, x direction.  
1 double: The receiver efficiency. Zero if fiber coupling is not computed.  
1 double: The system efficiency. Zero if fiber coupling is not computed.  
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.  
2\* $n_x$ \* $n_y$  double: Ex values.  
If polarized, 2\* $n_x$ \* $n_y$  Ey values follow the Ex values.

### ZEMAX Beam File (ZBF) ASCII format

The ZBF ASCII file format is defined as follows. The first line must be the single character "A", followed by the other data values specified.

A: indicates an ASCII file.  
version: The format version number, currently 0.  
 $n_x$ : The number of x samples.  
 $n_y$ : 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.



lambda: The wavelength in lens units of the beam in the current medium.  
 wx: The waist in lens units of the pilot beam, x direction.  
 re: The receiver efficiency. Zero if fiber coupling is not computed.  
 se: The system efficiency. Zero if fiber coupling is not computed.  
 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.  
 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\*n<sub>x</sub>\*n<sub>y</sub> Ex values.  
 If polarized, followed by 2\*n<sub>x</sub>\*n<sub>y</sub> Ey values.

### User defined DLL program

To define a beam using a 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. New DLLs may be easily created with a suitable compiler.

### Beam DLL parameters

Each DLL may use between zero and 4 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 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 \POP\DLL subdirectory off the main ZEMAX directory.**

## Surface specific settings

Each ZEMAX surface supports these settings relevant to Physical Optics Propagation:

**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. Multiple sequential surfaces may all have this option checked, in which case the beam will advance through all the surfaces using only ray propagation. Use of this feature is required for non-sequential, birefringent, and gradient index surfaces. 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.

**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.

**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 "beamfile name" 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:" on page 191 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.

**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- width is then set according to the equations described in "Comments about point spacing and sampling" on page 524. The number of points used in the X- and Y- sampling will not change.

**Reference Radius:** A rotationally symmetric beam has an identical phase radius of curvature in the X and Y directions. However, in the general case, there will be a difference between the X and Y phase radii. For cylinder or toroidal lenses, this difference may be extreme. The choice of reference radius determines how the pilot beam is propagated, which in turn affects the choice of propagation algorithm, array sampling, phase sampling, and other important data for determining the further propagation of the beam. Usually, the default "Best Fit" is the best choice; this selects the optimum trade-off between the X and Y radii to minimize the phase excursions relative to the reference surface. "Shorter" selects the shorter of the X or Y radius, "Longer" selects the longer of the two, and "X" and "Y" makes the choice explicitly. The "Plane" option will reference the phase to a plane, and this works well in cases where the deviation of the rays near the chief ray is a poor representation of the surface power, as is the case with axicons. The "User" option allows direct specification of the X and Y radius to use for the reference sphere. Use the value of zero to define a plane. Use "Best Fit" unless you are certain that another choice will yield higher accuracy for a given application. If "Separate X, Y" is selected on the beam definition tab, then the best fit radius in the X and Y directions is automatically selected, unless "Plane" or "User" is chosen for the reference radius. If the reference radius is set to "User" and "Separate X, Y" is not selected; the X radius value is used for both X and Y directions.

## **Computing Fiber Coupling**

The physical optics propagation algorithm may be used to compute fiber coupling efficiency. A ray based method is also supported, see "Fiber Coupling Efficiency" on page 174.

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 (see "Physical Optics" on page 187 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, see "Defining the initial beam" on page 526.

### 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" on page 190). 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 along the chief ray. The surface vertex option is not permitted when using the File option for defining the fiber mode.

## **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" on page 533.



***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" on page 529 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.

The probing ray set is an adequate model for the propagation near lenses. When propagating through surfaces, the probing ray set is used to determine the effective phase aberrations and change in overall reference sphere and pilot beam shape. The diffraction along the distance the probing rays trace is ignored. The probing ray set is not an accurate method of determining beam propagation near the focus of a ray bundle.

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.

## 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 \Samples\Physical Optics directory. 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  $\pi$  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.

## **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 macros can be stored on disk and called from within ZEMAX.

ZPL is similar to the BASIC programming language, except not all BASIC constructs and keywords are supported, and some new capabilities and functions unique to ray tracing have been added. If you are familiar with BASIC, you will learn ZPL quickly. If you are not, or if you have never programmed before, don't panic! ZPL is easy to use, and this chapter will give you instructions and examples to get you started.



***For a more powerful, compiled programming capability see "ZEMAX EXTENSIONS" on page 607.***

---

## **Creating ZPL macros**

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 chapter. Some example ZPL macros can also be found in the \MACROS directory, they are described in the chapter "ZEMAX example files".

Use any ASCII text editor to create the ZPL file (such as the NOTEPAD editor). Name the file any descriptive title you wish, but the file must end in the .ZPL extension.

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.

## **Running ZPL macros**

To run your ZPL macro, from the main menu, select Macros, Edit/Run ZPL Macros. 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 ASCII files ending in the extension .ZPL. The files must be in the directory for ZPL Macros; see the Chapter "File Menu" in the section on the Directories tab of the Preferences dialog box. Macros may also be placed in any subdirectory within the macros directory.

**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.

**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.

**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 the Windows NOTEPAD editor. The editor can be used to modify or rename the macro.

**View:** The view button will display the contents of the macro file in a text window which can be scrolled or printed. No editing is allowed in the view window.

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 statements or error messages will be placed in a text file. When the macro has terminated, the text file will be displayed in a window. The display of this output window can be suppressed using the CLOSEWINDOW keyword.

## Assigning ZPL macros to buttons

Frequently used macros may be assigned to buttons for convenient access. See "Buttons 1-16, Buttons 17-32, Buttons 33-48" on page 63 for details.

## An overview of ZPL

A ZPL macro consists of a series of commands (called statements) which are stored in an ASCII text file. You can create a ZPL macro with any text editor outside of ZEMAX (you can also edit the macro from within ZEMAX; this will be discussed in a subsequent section). The ZPL language uses a syntax similar to BASIC for most (but not all) commands. For example, the statement

```
x = 5
```

is one type of valid ZPL statement. There are several important things to note in this statement. 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. There is no need to declare the variable before using it. Second, no special symbol is required to terminate a statement, such as ";" in C. Because of this, each ZPL statement must be on a line by itself.

ZPL supports "inline" function calls. This allows statements of the following forms to be used:

```
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 single-operand functions, all of which are defined in a subsequent section. 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.

All of the previously shown statements have one thing in common: they are all assignments. This means the expression on the right side of the equals sign is evaluated, and the result is assigned to the variable designated on the left. There is one other kind of statement in ZPL, and it is called a keyword.

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 statements

```
x = 3
y = 4
z = SQRT(x * x + y * y)
PRINT "The hypotenuse is ", z
```

will print the following to the screen:

```
The hypotenuse is 5.0000
```

Note that ZPL enforces operator precedence. Within any pair of parentheses, operations are carried out in a particular order. ZPL uses the following precedence from highest to lowest: functions (such as SQRT), logical operators (such as ==), multiplication and division, and then addition and subtraction. Parentheses always override precedence, in this sense they have the highest priority.

There are many other keywords which will be described in a subsequent section.

## ZPL 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 statement

```
x=5
```

presented earlier 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 however 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.

Violating these rules may generate a syntax error, however the macro may appear to run without errors and will not compute what you really want. All ZPL variables are stored internally as 64-bit double precision numbers.

## **ZPL functions**

ZPL functions can be used on the right hand side of an 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.

Many functions have corresponding keywords. For example, RADI() is a function which returns the radius of the surface corresponding to the argument (RADI(3) returns the radius of surface 3). There is also a keyword called RADI which is used to set the radius of that surface. See the keyword descriptions for more information.

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.
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.
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.

Function	Argument	Return value
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" on page 546
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.
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" on page 93.
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.

Function	Argument	Return value
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 subsequently be used by GLAS 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".
GPARG(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.
IMAE()	Image analysis efficiency.	Returns the geometric image analysis efficiency. See the discussion "Optimizing with the IMAE operand" on page 428.
INDX(surf)	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(surf)	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.
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.

Function	Argument	Return value
MCON(row, config, data)	Row (operand number), Configuration number, and data value to extract from the Multi-Configuration Editor.	<p>Extracts data from any row and configuration of the Multi-Configuration Editor. This function is similar to MCOP with extended capabilities for extracting data.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>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\$:</p> <pre>dummy = MCON(1, 0, 0) a\$ = \$buffer()</pre> <p>See also keyword SETMCOOPERAND.</p>
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 also keyword SETMCOOPERAND.
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(surf)	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" on page 590.
NPOS(surf, object, code)	surf is the surface number, object is the object number, and code is 0-6 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" on page 591.

Function	Argument	Return value
NPRO(surf, object, code, csg)	surf is the surface number, object is the object number, code values are as defined for the SETNSCPROPERTY keyword, and csg is the coating and scattering group 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" on page 591. 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 desired detector. Pixel indicates which pixel value is returned. Data is 0 for real, 1 for imaginary, 2 for the amplitude sum, and 3 for the coherent intensity. See also "NSTR" on page 575.	If the object number corresponds to a detector object, then the coherent intensity data from the specified pixel is returned. Coherent intensity consists of 4 numbers which are each summed over all incident rays: the real part, the imaginary part, the amplitude, and the incoherent intensity.
NSDD(surf, object, pixel, data)	The surf value indicates the surface number of non-sequential group. The object number indicates the desired detector. Pixel indicates which pixel value is returned, see the discussion to the right for details. Data is 0 for flux, 1 for flux/area, and 2 for flux/solid angle pixel. See also "NSTR" on page 575.	If the object number is zero, then all detectors are cleared and the function returns zero. If the object number corresponds to a detector object, then the incoherent intensity data from the specified pixel is returned. If the pixel number is zero, then the sum of the flux or average flux/area for all pixels for that detector object is returned. If the pixel number is -1, then the maximum flux or flux/area is returned. If the pixel number is -2, then the minimum flux or flux/area is returned. If the pixel number is -3, the number of rays striking the detector is returned. If the pixel number is -4, the RMS deviation from the mean is returned. If data is 2 then the pixel number must be greater than 0, otherwise, a value of 0 is returned. Only data values 0 and 1 (for flux and flux/area) are supported for faceted detectors.
NSUR()	(null)	The number of defined surfaces.
NWAV()	(null)	The number of defined wavelengths.
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.

Function	Argument	Return value
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 or RAYTRACEX call. OPDC will not return valid data if the chief ray cannot be traced.
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 Hx-Py, 8 for target, 9 for weight, 10 for value, and 11 for percent contribution. 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, hx, hy, px, py)	Code is the optimization operand code (see function OCOD), and int1, int2, hx, hy, px, and py values for the operand. See "Optimization operands" on page 391	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.
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 or RAYTRACEX call. OPTH will not return valid data if the chief ray cannot be traced.
PARM(n,s)	Valid parameter number and surface number	Parameter "n" of surface "s".
PARn(x)	Valid surface number	Obsolete, use PARM instead.
PMOD()	(null)	0 if paraxial mode is off, else 1.
POWR(x,y)	x is positive, y is any number	Computes x to the power of y. X must be positive, or the absolute value is used.
PVHX()	(null)	"Hx" parameter from the ZPLM optimization operand.
PVHY()	(null)	"Hy" parameter from the ZPLM optimization operand.
PVPX()	(null)	"Px" parameter from the ZPLM optimization operand.

Function	Argument	Return value
PVPY()	(null)	"Py" 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.
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.
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.

Function	Argument	Return value
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.
SCOD(A\$)	Any string variable or literal that is the name of a ZEMAX surface type. The valid names are 8 character strings displayed on the prescription data report for each surface type in use. These are not the same names that are displayed in the Lens Data Editor.	The surface type code of the surface. See function STYP and "SURFTYPE" on page 598.
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\$
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 semi-diameter, and 5 for TCE. Param is an integer between 0 and 3.	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 the solve parameters. For more information on the solve data and the meaning of the solve parameters, see "SOLVES" on page 379.



Function	Argument	Return value
SPRO(surf, code)	Surf is the surface number and code values are as defined for the SETSURFACEPROPERTY keyword.	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" on page 593. 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.
SQRT(x)	Positive numeric expression	Square root of the expression.
STDD(surf, parm)	Surface number, tilt and decenter data parameter	The value of the specified surface tilt and decenter parameter. For a description of the parameter numbers, see "SETSTDD" on page 593.
STYP(n)	Surface number n	The surface type code of the surface. See function SCOD and "SURFTYPE" on page 598
SVAL(A\$)	Any string variable name	String value. Returns a floating point value of the string A\$.
TANG(x)	Numeric expression in radians	Tangent of the expression.
TMAS()	Total mass in grams	The total mass in grams of the lens from surface 1 to the image surface.
THIC(x)	Valid surface number	Thickness of the surface.
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 four digit version code. A typical version number is 10000 for version 10.0. Subsequent versions of ZEMAX will return larger version codes.
WAVL(x)	Valid wavelength number	Wavelength in micrometers.
WWGT(x)	Valid wavelength number	Wavelength weighting.
XMIN()	(null)	The minimum x-coordinate in the graphics window.

Function	Argument	Return value
XMAX()	(null)	The maximum x-coordinate in the graphics window.
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)

```

The fiber coupling is computed by calling FICL(n) where n is the vector number containing the argument list.

## **ZPL math operations**

ZPL macros can include basic math 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 ZPL functions or ZPL logical operators, both described in subsequent sections.

## **ZPL logical operators**

Logical operators are used to construct complex statements which ultimately evaluate to one or zero. Most logical operations take the form (left\_expression) (operator) (right\_expression), similar to mathematical statements 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 statements 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 statements. For example, an IF statement may contain two conditions which must both be true for the THEN statement 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.

### ZPL 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.

### **ZPL string variables and operations**

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 statement 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. 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$
```

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. The string functions can be used in a defining statement 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 statements. 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$ = $LENSNAME( )
PRINT Z$
```

One very important function is \$STR(). This function accepts as an argument any expression which evaluates to a number. \$STR is useful for formatting numeric data into strings:

```
A$ = "The expression evaluates to " + $STR( SQRT(3*3 + 4*4 + z*y) )
```

The inverse function is SVAL(A\$), which converts a string into a floating point value.

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.
\$COAT(i)	Returns the coating name for the ith surface.
\$COMMENT(i)	Returns the comment string for the ith surface.
\$DATE()	Returns the current date and time string. The formatting is specified by the Date/Time control settings described under "Miscellaneous" on page 60.
\$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\$. For example, if A\$ = "one two three", then \$GETSTRING(A\$, 2) returns "two". Spaces and commas are used as delimiters, although commas count as separate sub-strings. See \$LEFTSTRING and \$RIGHTSTRING.
\$GLASS(i)	Returns the glass name of surface number i.
\$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. See \$GETSTRING and \$RIGHTSTRING.
\$LENSNAME()	Returns the lens title defined in the General System dialog box.
\$NOTE(line#)	<p>Returns the notes information defined in the General System dialog box. 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.</p> <p>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 General dialog however). \$NOTE will return a null (empty) string if there are no defined characters in the notes for the specified line.</p> <p>The function SLEN can be used to determine the actual number of characters returned by \$NOTE using this syntax:</p> <pre>A\$ = \$NOTE(1) N = SLEN(A\$)</pre> <p>The value N will be the integer number of characters in the string A\$.</p>
\$OBJECTPATH()	Returns the path name for NSC object files.

Function	Description
\$PATHNAME()	Returns the path name only for the current lens file. This is useful for determining the directory where the lens file is stored.
\$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. See \$GETSTRING and \$LEFTSTRING.
\$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.
\$TEMPFILENAME	Returns the name of a temporary file, with complete path, suitable for temporary storage of text or binary data. See keyword GETTEXTFILE.
\$UNITS()	Returns either MM, CM, IN, or M, depending upon the current lens units.

## **ZPL 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.

ZPL 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 statement may compare strings as follows:

```
A$ = "TEST"
BS = "TEST"
IF (A$ $== B$) THEN PRINT "Strings are identical."
```

## **ZPL keywords**

ZPL 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. Certain keywords with similar or related functions are grouped together.

### **APMN, APMX**

#### *Purpose:*

Used to set the aperture minimum/maximum radius of a surface.

#### *Syntax:*

```
APMN(surface) = (new_value)
APMX(surface) = (new_value)
```

#### *Discussion:*

These keywords requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored.

Otherwise, the expression (new\_value) is evaluated, and assigned to the aperture minimum (APMN) or aperture maximum (APMX) radius.

The aperture minimum/maximum radius data is used by all of the aperture types, although the meaning of the data depends upon which aperture is being used, as described in the following table.

**APERTURE TYPES AND APMN USAGE**

Aperture type	APMN is used for:	APMX is used for:
None.	Nothing	Nothing
Circular aperture	Setting the minimum aperture radius	Setting the maximum aperture radius
Circular obscuration	Setting the minimum aperture radius	Setting the maximum aperture radius
Spider	Setting the width of each spider arm	Setting the number of spider arm
Rectangular aperture	Setting the x-half width	Setting the y-half width
Rectangular obscuration	Setting the x-half width	Setting the y-half width
Elliptical aperture	Setting the x-half width	Setting the y-half width
Elliptical obscuration	Setting the x-half width	Setting the y-half width
User aperture	Nothing	Nothing
User obscuration	Nothing	Nothing
Floating aperture	Nothing	Nothing

*Example:*

```
APMN(3) = 1.75
APMX(3) = 3.50
```

### **APXD, APYD**

*Purpose:*

Used to set the aperture x/y decenter of a surface.

*Syntax:*

```
APXD(surface) = (new_value)
APYD(surface) = (new_value)
```

*Discussion:*

These keywords requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored. Otherwise, the expression (new\_value) is evaluated, and assigned to the aperture x-decenter (APXD) or aperture y-decenter (APYD).

*Example:*

```
APXD(3) = -3.6
APYD(5) = -1 * APYD(3)
```

### **APTP**

*Purpose:*

Used to set the aperture type of a surface.

*Syntax:*

APTP(surface) = integer\_code

**Discussion:**

This keyword requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored. Otherwise, the expression (new\_value) is evaluated, rounded to an integer and assigned to the aperture type.

The aperture type is stored as an integer code described in the following table.

**APERTURE TYPE CODES**

Integer code	Aperture type
0	None.
1	Circular aperture
2	Circular obscuration
3	Spider
4	Rectangular aperture
5	Rectangular obscuration
6	Elliptical aperture
7	Elliptical obscuration
8	User aperture
9	User obscuration
10	Floating aperture

**Example:**

APTP(3) = 1

**ATYP, AVAL**

**Purpose:**

Used to set the system aperture type and value.

**Syntax:**

ATYP = (code)  
AVAL = (new\_value)

**Discussion:**

These keywords are used to define the system aperture type and value. The UPDATE command must be issued before the new data takes effect. The aperture type is defined using one of the following codes:

**SYSTEM APERTURE TYPE CODES**

Aperture type	Code
Entrance Pupil Diameter	0
Image Space F/#	1
Object Space Numerical Aperture	2
Float By Stop Size	3

### *Example:*

```
! Set the EPD to be 35
ATYP = 0
AVAL = 35.0
```

## **BEEP**

### *Purpose:*

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.

## **COAT**

### *Purpose:*

COAT is used to set the name of the coating on a surface.

### *Syntax:*

```
COAT (surface), coatingname
```

### *Discussion:*

This keyword requires the surface number, and the name of the new coating to apply. Leave the name blank if the surface is to be uncoated.

### *Example:*

```
COAT 2, WAR
A$ = "AR"
COAT 3, A$
```

### *Related Keywords:*

UPDATE

## **CLOSE**

### *Purpose:*

Closes an ASCII file previously opened by the OPEN command.

### *Syntax:*

```
CLOSE
```

### *Discussion:*

See the description for OPEN.

## **CLOSEWINDOW**

### *Purpose:*

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

### *Purpose:*

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. If the argument n is zero, the color will be set to black. Otherwise, the color defined by the integer value will be used for all subsequent line and text commands in graphics mode. There are 12 valid colors plus black; numbered 0-12. The colors are normally used by ZEMAX for fields and wavelengths; color 1 is for either field 1 or wavelength 1, etc.

## COMMENT

### *Purpose:*

COMMENT is used to set the comment of a surface.

### *Syntax:*

```
COMMENT (surface) = "literal string"  
COMMENT (surface) = stringvariable$
```

### *Discussion:*

This keyword requires two expressions, one to specify the surface number and the other to define the comment string. COMMENT also accepts a string variable which contains the comment.

### *Example:*

```
COMMENT 4 = "Comment for surface 4"  
COMMENT 6 = A$
```

### *Related Keywords:*

UPDATE

## CONI

### *Purpose:*

CONI is used to set the conic constant of a surface.

### *Syntax:*

```
CONI (surface) = (new_value)
```

### *Discussion:*

This keyword requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored. Otherwise, the expression (new\_value) is evaluated and assigned.

### *Example:*

```
CONI 1 = -1
```

### *Related Keywords:*

UPDATE

## COPYFILE

### *Purpose:*

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**

*Purpose:*

CURV is used to set the curvature of a surface.

*Syntax:*

```
CURV (surface) = (new_value)
```

*Discussion:*

This keyword requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored. Otherwise, the expression (new\_value) is evaluated and assigned.

*Example:*

```
CURV(3) = - CURV(4) + .001
```

*Related Keywords:*

UPDATE

## **DELETE**

*Purpose:*

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**

*Purpose:*

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**

*Purpose:*

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

*Purpose:*

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

*Purpose:*

DELETEMFO deletes an existing operand in the merit function editor.

*Syntax:*

DELETEMFO (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 INSERTMFO.

## DELETEOBJECT

*Purpose:*

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

## EDVA

*Purpose:*

Sets the extra data values for a surface.

*Syntax:*

EDVA surf, ed\_value, new\_value

*Discussion:*

The surf expression indicates which surface is to be modified. The ed\_value expression must evaluate to an integer corresponding to one of the extra data value positions. See the chapter "Surface types" for information on the extra data values. The new\_value expression is evaluated and is used for the new value of the extra data value on that surface.

*Example:*

```
EDVA 5, 6, x+y
```

**END**

See GOSUB.

**EXPORTBMP**

*Purpose:*

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 graphic 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 allow the graphic to be completely redrawn and the screen capture to complete. If the JPG files appear incomplete, try a delay value of 500 - 2500 milliseconds. See also EXPORTJPG and EXPORTWMF.

*Example:*

```
EXPORTBMP 1, "C:\TEMP\MYBMPPFILE"  
EXPORTBMP k, A$
```

**EXPORTCAD**

*Purpose:*

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 IGES/SAT/STEP Solid" on page 208. 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).

VEC1(3): The First surface to export.

VEC1(4): The last surface 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): The ray pattern. Use 0 for XY, 1 for X, 2 for Y, 3 for ring, 4 for list, 5 for none.

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 - n 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 IGES/SAT/STEP Solid" on page 208.

*Example:*

```
VEC1(1) = 0
VEC1(2) = 32
VEC1(3) = 1
etc...
VEC1(18) = 0
EXPORTCAD "C:\TEMP\MYCADFILE.IGS"
```

## **EXPORTJPG**

*Purpose:*

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
```

## **EXPORTWMF**

*Purpose:*

Exports any graphic window as a Windows Metafile. Unlike EXPORTBMP and EXPORTJPG, EXPORTWMF requires the filename to have the desired extension included in the filename argument. The metafile format will be whatever the current selected metafile type is, see "Graphics" on page 59.

*Syntax:*

```
EXPORTWMF winnum, filename
```

*Discussion:*

See EXPORTBMP.

*Example:*

```
EXPORTWMF 3, "C:\TEMP\MYWMFFILE.EMF"
EXPORTWMF k, FILENAME$
```

## **FINDFILE**

*Purpose:*

Used to find the names of files stored on disk.

*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 directory, 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**

*Purpose:*

FLDX and FLDY are used to change the field point definitions. FWGT is used to change the field weight. FVDX and FVDY change the x and y vignetting decenter factors. FVCX and FVCY change the x and y vignetting compression factors. FVAN changes the vignetting angle.

*Syntax:*

```
FLDX (field number) = (new_value)
FLDY (field number) = (new_value)
FWGT (field number) = (new_value)
FVDX (field number) = (new_value)
FVDY (field number) = (new_value)
FVCX (field number) = (new_value)
FVCY (field number) = (new_value)
FVAN (field number) = (new_value)
```

*Discussion:*

These keywords require two expressions, one to specify the field number and the other to define the new value. The expression (field number) is evaluated and then rounded down to an integer to yield the field number. If the field number is less than 1 or greater than the number of fields, the command is ignored. Otherwise, the expression (new\_value) is evaluated and assigned. The UPDATE command must be issued before the new data takes effect.

*Example:*

```
FLDX 1 = 0.0
FLDY 1 = 12.5
FWGT 1 = 1.0
```

*Related Functions:*

FLDX, FLDY, FWGT, MAXF, FTYP, FVDX, FVDY, FVCX, FVCY, FVAN

*Related Keywords:*

UPDATE, FTYP, NUMFIELD

**FOR, NEXT**

*Purpose:*

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
    statements...
```

NEXT

*Discussion:*

The keyword FOR marks the beginning of a group of statements 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 statements. FOR-NEXT loops may be nested. The number of FOR and NEXT statements must be the same.

Upon reaching a FOR statement, 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.

*Example:*

```
FOR i = 1, 25, 1
    PRINT i
NEXT

j = 5
k = 0
FOR i = j, j + 5, 2
    k = i + j + k
NEXT
```

## FORMAT

*Purpose:*

Specifies the numerical precision format for subsequent PRINT 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 PRINT statements 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 statement will be ignored. The optional keyword EXP after the m.n expression indicates exponential notation should be used.

*Example:*

```
x = 1.123
FORMAT 12.0
PRINT "An integer portion=", x
FORMAT 12.8
PRINT "A decimal portion = ", x
FORMAT 12.8 EXP
PRINT "Exponential notation = ", x
```

## FTYP

*Purpose:*

FTYP is used to change the type of field definition between angle in degrees, object height, and image height.

*Syntax:*

```
FTYP = (new_value)
```

*Discussion:*

The (new value) is an expression that must evaluate to 0, 1, 2, or 3. Use 0 to set the field type to angle in degrees, 1 for object height in lens units, 2 for paraxial image height in lens units, 3 for real image height in lens units. Setting the field type does not change the field values, nor is the system updated. Usually, the field values must be adjusted with the FLDX and FLDY keywords whenever the field type is changed. The UPDATE command must be issued before the new data takes effect.

*Related Functions:*

FLDX, FLDY, FWGT, MAXF

*Related Keywords:*

UPDATE, FLDX, FLDY, FWGT

## GCRS

*Purpose:*

GCRS is used to change the Global Coordinate Reference Surface. See "Global Coordinate Reference Surface" on page 93.

*Syntax:*

GCRS = (new\_value)

*Discussion:*

The (new value) is an expression that must evaluate to a valid surface number.

## GDATE

*Purpose:*

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 your graphics look like other ZEMAX graphics.

*Example:*

See the section GRAPHICS.

## GETEXTRADATA

*Purpose:*

Retrieves the extra data values from the extra 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 the chapter "Surface Types" for descriptions of the extra data values.

## GETGLASSDATA

*Purpose:*

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: 1 for Schott, 2 for Sellmeier 1,
        3 for Herzberger, 4 for Sellmeier 2, 5 for Conrady.
2:      MILNUM
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 (meaning depends upon formula)
17-22: Thermal constants of dispersion
23+: 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.
```

**Related Functions:**

GNUM

**GETLSF**

**Purpose:**

Calculates the geometric edge and line response functions, similar to the “Geometric Line/Edge Spread” on page 137.

**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**

### *Purpose:*

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 line pairs per millimeter. 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" on page 114).

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**

### *Purpose:*

Calculates the diffraction point spread function (PSF) data for the currently loaded lens file, 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.

Vector position 1 through  $4*n*n$  holds the PSF data intensity, normalized to unity. 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

### *Purpose:*

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. 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:      Use Env Data (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:     Use Ray Aiming (1 if true, 0 if false)
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 flag (0 for false, 1 for true)
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
```

## **GETTEXTFILE**

### **Purpose:**

Creates an ASCII 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 case-sensitive label that indicates the type of analysis to be performed. The 3 letter labels are identical to those used for the button bar in ZEMAX. A list of codes may be found on the "Buttons" tab of the File, Preferences dialog box. The labels are case sensitive. If no label 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.

Only text, and not graphic files, are supported by GETTEXTFILE.

See also OPEN, CLOSE, READ, READNEXT, and READSTRING.

#### *Example:*

```
! Macro sample to generate cardinal points in ASCII 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$
ABEL 1
READSTRING B$
IF (EOFF()) THEN GOTO 2
PRINT B$
GOTO 1
LABEL 2
CLOSE
```

## GETVARDATA

#### *Purpose:*

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

#### *Purpose:*

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" on page 165, "Zernike Standard Coefficients" on page 168, and "Zernike Annular Coefficients" on page 171 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

**Purpose:**

GLAS is used to set the glass type of a surface.

**Syntax:**

```
GLAS (surface) = (glass_number)
GLAS (surface) = glass$
```

**Discussion:**

This keyword requires two expressions, one to specify the surface number and the other to define the number of the glass. If the glass number is -1, the glass type is set to "mirror". If the glass number is 0, then the glass type is blank, or unity index. Otherwise, the glass number must be between 1 and the number returned by MAXG. GLAS also accepts a string variable which contains the name of the desired glass, such as "BK7".

**Example:**

```
GLAS 3 = MAXG()
A$ = "BK7"
GLAS 4 = GNUM(A$)
GLAS 4 = A$
```

**Related Keywords:**

UPDATE

## GLENSNAME

**Purpose:**

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.

## GOSUB, SUB, RETURN, and END

*Purpose:*

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 statement. More than one return statement may be placed in the body of a subroutine. If subroutines are defined, at least one END statement 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
```

## GOTO

*Purpose:*

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

*Purpose:*

Creates a standard ZEMAX graphics frame with ruling lines for the plot title.



### *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

## **GTEXT**

### *Purpose:*

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 the section GRAPHICS.

## **GTEXTCENT**

### *Purpose:*

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 the section GRAPHICS.

## **GTITLE**

*Purpose:*

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 your ZPL graphics functions look like standard ZEMAX graphics.

*Syntax:*

```
GTITLE user_title
```

*Discussion:*

GTITLE is primarily used for making your graphics look like other ZEMAX graphics.

*Example:*

See the section GRAPHICS.

## **HAMMER**

*Purpose:*

Invokes the Hammer optimization algorithm to optimize the current lens with the current merit function.

*Syntax:*

```
HAMMER  
HAMMER number_of_cycles
```

*Discussion:*

If no argument is provided, then the hammer optimization runs 1 cycle. 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. Currently, the ZPL implementation of this function does not take advantage of multiple processors.

*Related Functions:*

MFCN

*Example:*

```
PRINT "Starting merit function:", MFCN()  
HAMMER 3  
PRINT "Ending merit function :", MFCN()
```

## **IF-THEN-ELSE-ENDIF**

*Purpose:*

IF provides conditional macro execution and branching.

*Syntax:*

```
IF (expression)  
(statements)  
ELSE  
(statements)  
ENDIF  
  
or
```

```
IF (expression) THEN (statement)
```

*Discussion:*

The IF-ELSE-ENDIF construction is used for conditional execution of either the group of statements following the IF statement or the statements following the ELSE statement, 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 statement 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 statement 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

*Purpose:*

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” on page 144. For a description of the Geometric Image Analysis feature see “Geometric Image Analysis” on page 142.

*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 \ImaFiles subdirectory. The infilename must also be placed in the \ImaFiles subdirectory. 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

## IMASHOW

*Purpose:*

Displays an IMA or BIM file in a viewer window. For a description of the IMA and BIM file formats, see “The IMA format” on page 144 and “The BIM format” on page 144.

*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 \ImaFiles subdirectory. This command will open a new window to display the file.

*Example:*

```
IMASHOW "LETTERF.IMA"
```

*Related Keywords:*

IMA, IMASUM

## IMASUM

### *Purpose:*

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" on page 144 and "The BIM format" on page 144.

### *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 \ImaFiles subdirectory, 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

## IMPORT ED

### *Purpose:*

Imports data into the extra data editor from a file.

### *Syntax:*

```
IMPORT_ED surface filename
```

### *Discussion:*

The surface may be any valid surface number that uses extra data values. The data to be imported should be a single column of numeric data in ASCII format. See "Tools" under "Extra Data" on page 79, and "Importing grid data" on page 259.

## INPUT

### *Purpose:*

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.

### *Example:*

```
INPUT "Enter value for x:", x  
PRINT "X = ", x  
INPUT "Enter a value for A$:", A$  
PRINT A$
```

## INSERT

### *Purpose:*

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**

**Purpose:**

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**

**Purpose:**

INSERTMCO inserts a new multi-configuration operand in the multi-configuration editor.

**Syntax:**

```
INSERTMCO (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 DELETETMCO and INSERTCONFIG.

**Example:**

```
INSERTMCO 7
```

## **INSERTMFO**

**Purpose:**

INSERTMFO inserts a new merit function operand in the merit function editor.

**Syntax:**

```
INSERTMFO (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 DELETETMFO and SETOPERAND.

**Example:**

```
INSERTMFO 23
```

## **INSERTOBJECT**

**Purpose:**

Inserts a new NSC object.

**Syntax:**

```
INSERTOBJECT (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 new object will be inserted at the specified object number, renumbering other objects as required. See also DELETEOBJECT and SETNSCPROPERTY.

*Example:*

```
INSERTOBJECT 1, 1
```

## **LABEL**

*Purpose:*

LABEL provides a destination for the GOTO command, see the section "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 statements in a macro.

*Example:*

```
LABEL 7  
LABEL startover
```

## **LINE**

*Purpose:*

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 the section GRAPHICS.

## **LOADCATALOG**

*Purpose:*

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.

## **LOADLENS**

*Purpose:*

Loads a new lens file from disk, replacing any lens file currently in memory.

**Syntax:**

```
LOADLENS "filename" [appendflag, session]  
LOADLENS file$ [appendflag, session]
```

**Discussion:**

LOADLENS will load a new lens file from disk. 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 directory for lenses will be used (see the File menu chapter under Environment).

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. 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**

**Purpose:**

Loads a merit function file from disk, replacing the merit function in the current lens.

**Syntax:**

```
LOADMERIT "filename"  
LOADMERIT file$
```

**Discussion:**

LOADMERIT will load a new merit function from disk. 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 default directory for lenses will be used (see "Directories" on page 59). See also SAVEMERIT.

## **LOCKWINDOW**

**Purpose:**

Locks any open window.

**Syntax:**

```
LOCKWINDOW winnum
```

**Discussion:**

See "Graphic windows operations" on page 36.

**Example:**

```
LOCKWINDOW 7
```

**Related Keywords:**

UNLOCKWINDOW

## **NEXT**

See FOR.

## **NSTR**

**Purpose:**

Initiates a Non-sequential trace.

**Syntax:**

```
NSTR surf, source, split, scatter, usepolar, ignore_errors, no_random_seed, save,  
savefilename, filter
```

### *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 no\_random\_seed is zero, then every call to NSTR will use different random rays, otherwise, the rays will be the same every time, which is required when using NSTR for optimization.

If save is omitted or is zero, the arguments savefilename and filter need not be supplied. If save is not zero, the rays will be saved in a ZRD file. The ZRD file will have the name specified by the savefilename, and will be placed in the same directory as the lens file. The extension of savefilename should be ZRD, and 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. For information on filter strings see "The filter string" on page 371.

### *Related Functions:*

NSDD

### *Example:*

```
NSTR 1, 0, 0, 0, 0, 1, 0, 1, "saverays.ZRD", "h2"
```

## **NUMFIELD**

### *Purpose:*

Sets the total number of defined fields. The UPDATE command must be issued before the new data takes effect.

### *Syntax:*

```
NUMFIELD <expression>
```

## **NUMWAVE**

### *Purpose:*

Sets the total number of defined wavelengths. The UPDATE command must be issued before the new data takes effect.

### *Syntax:*

```
NUMWAVE <expression>
```

## **OPEN**

### *Purpose:*

Opens an existing ASCII numeric 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
```

## **OPTIMIZE**

### ***Purpose:***

Invokes the optimization algorithm to optimize the current lens with the current merit function.

### ***Syntax:***

```
OPTIMIZE
OPTIMIZE number_of_cycles
```

### ***Discussion:***

If an argument is provided, it must evaluate to an integer value between 1 and 99, and the optimization algorithm will run the specified number of cycles. If no argument is provided, or number\_of\_cycles evaluates to zero, then the optimization runs in "Automatic" mode, stopping when the algorithm detects the process has converged. 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**

### ***Purpose:***

Used to return numerical values back to the optimization algorithm through the use of the ZPLM optimization operand.

### ***Syntax:***

```
OPTRETURN (datafield) = (result)
```

### ***Discussion:***

OPTRETURN takes two arguments, separated by an equal sign. 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 the chapter "Optimization" for details.

### ***Example:***

```
x = sqrt(thic(3) + radi(5))
OPTRETURN j = x+5
```

## **OUTPUT**

### ***Purpose:***

Specifies destination for text output. Output is either to the screen, or to a file.

### ***Syntax:***

```
OUTPUT SCREEN
OUTPUT "filename"
OUTPUT "filename" APPEND
OUTPUT A$
OUTPUT A$ APPEND
```

### ***Discussion:***

If OUTPUT SCREEN is specified alone, then all subsequently executed PRINT statements will be directed to the screen. If a valid filename is provided, then subsequent PRINT statements will output to the filename specified. 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.

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

*Purpose:*

PARM is used to set parameter values on a surface. The parameter values are defined in the chapter "Surface Types".

*Syntax:*

```
PARM (number), (surface), (new_value)
```

*Discussion:*

This keyword requires three expressions: the parameter number, surface number, and the value.

*Example:*

```
PARM 2, 4, 250
```

*Related Keywords:*

UPDATE

## PARn

*PARn is obsolete, use PARM instead.*

## PARAXIAL

*Purpose:*

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 = PMODE()
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

*Purpose:*

Pauses macro execution 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 x
PAUSE "Ready to continue..."
PAUSE x+sqrt(5)
```

**Discussion:**

This feature may be used for debugging, presenting results, or pausing the execution of the macro.

## **PIXEL**

**Purpose:**

Turns on a single pixel on the current graphics screen.

**Syntax:**

```
PIXEL xcoord, ycoord
```

**Discussion:**

This feature is useful for making spot diagrams.

## **POLDEFINE**

**Purpose:**

Defines the input polarization state for subsequent POLTRACE calls.

**Syntax:**

```
POLDEFINE Ex, Ey, PhaX, PhaY
```

**Discussion:**

The POLDEFINE keyword is used to define the input polarization state for subsequent polarization ray tracing. POLDEFINE requires the normalized Ex and Ey electric field magnitudes, as well as the X and Y phase angles in degrees. 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**

**Purpose:**

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 amplitude reflection, real
9:      S-Polarization amplitude reflection, imaginary
10:     S-Polarization amplitude transmission, real
11:     S-Polarization amplitude transmission, imaginary
12:     P-Polarization amplitude reflection, real
13:     P-Polarization amplitude reflection, imaginary
14:     P-Polarization amplitude transmission, real
15:     P-Polarization 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

```

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**

#### *Purpose:*

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” on page 190. For information on the ZBF file format, see “ZEMAX Beam File (ZBF) binary format” on page 528.

#### *Syntax:*

POP outfilename, lastsurface

#### *Discussion:*

This keyword requires the name of the output ZBF file, and an expression that evaluates to the last surface to propagate to. 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\Beamfiles subdirectory. 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. To make adjustments to the settings, open a POP window, choose the appropriate settings, then press "Save". 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
```

## **PRINT**

#### *Purpose:*

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**

### *Purpose:*

Prints a text file.

### *Syntax:*

```
PRINTFILE "filename"
PRINTFILE NAME$
```

### *Discussion:*

The filename must be a valid filename within quotes or the name of a string variable which contains a valid file name. The file must be an ASCII file (as would be created by OUTPUT and PRINT statements in ZPL) and must be in the current directory. PRINTFILE also closes the file if no CLOSE statement has been executed.

### *Example:*

```
OUTPUT "test.txt"
PRINT "Print this to the printer."
PRINTFILE "test.txt"
```

### *Related Keywords:*

OPEN, OUTPUT, CLOSE, PRINT, PRINTFILE

## **PRINTWINDOW**

### *Purpose:*

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**

### *Purpose:*

Sets the primary wavelength.

### *Syntax:*

PWAV (wavelength number)

### *Discussion:*

The expression (wavelength number) 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**

### *Purpose:*

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**

### *Purpose:*

RADI is used to set the radius of curvature of a surface.

### *Syntax:*

```
RADI (surface) = (new_value)
```

### *Discussion:*

This keyword requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored. Otherwise, the expression (new\_value) is evaluated and assigned.

### *Example:*

```
RADI i+m = -1200
```

### *Related Keywords:*

UPDATE

## **RANDOMIZE**

### *Purpose:*

RANDOMIZE seeds the random number generator.

### *Syntax:*

```
RANDOMIZE (seed)
```

### *Discussion:*

If seed evaluates to zero or a negative number, 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

## **RAYTRACE**

### *Purpose:*

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 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 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. 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**

### *Purpose:*

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. 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**

**Purpose:**

Reads data from an existing ASCII numeric 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 ASCII 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 statement 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.

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, READSTRING



## READNEXT

### *Purpose:*

Reads data from an existing ASCII numeric 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.

### *Example:*

```
OPEN "C:\DATA\TEST.DAT"
READNEXT x1, x2
READNEXT x3
CLOSE
```

### *Related Keywords:*

OPEN, CLOSE, READ, READSTRING

## READSTRING

### *Purpose:*

Reads data from an existing ASCII text file opened for reading by the OPEN command.

### *Syntax:*

```
READSTRING A$
```

### *Discussion:*

The ASCII 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. 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

## RELOADOBJECTS

### *Purpose:*

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, !

*Purpose:*

REM is used to indicate the rest of the line is a remark.

*Syntax:*

```
REM text  
! 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, prior to any spaces, tabs, or other characters. Any other use will cause a syntax error at run time.

*Example:*

```
REM any text can be placed after the REM command.  
! any text can also be placed  
! after the exclamation symbol.
```

## REMOVEVARIABLES

*Purpose:*

Sets all currently defined variables to fixed status.

*Syntax:*

```
REMOVEVARIABLES
```

## RENAMEFILE

*Purpose:*

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

*Purpose:*

REWIND erases the last line printed by the PRINT statement, 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

## **SAVELENS**

*Purpose:*

Saves the current lens file.

*Syntax:*

```
SAVELENS ["filename"]
SAVELENS NEW$
```

*Discussion:*

SAVELENS will save the current lens file to disk. 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.

*Example:*

```
SAVELENS
SAVELENS "NEWCOPY.ZMX"
SAVELENS NEW$
```

*Related Keywords:*

LOADLENS

## **SAVEMERIT**

*Purpose:*

Saves the current merit function to a file.

*Syntax:*

```
SAVEMERIT "filename"
SAVEMERIT file$
```

*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 default directory for lenses will be used (see "Directories" on page 59). See also LOADMERIT.

## **SAVEWINDOW**

*Purpose:*

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**

### *Purpose:*

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**

### *Purpose:*

SDIA is used to set the semi-diameter of a surface.

### *Syntax:*

```
SDIA (surface) = (new_value)
```

### *Discussion:*

This keyword requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored. Otherwise, the expression (new\_value) is evaluated and assigned.

If new\_value evaluates to a positive number, then the semi-diameter status will be changed to "Fixed", no matter what the initial status was. If new\_value is negative, then the status is set to "Automatic" which will be updated the next time "UPDATE" is called.

### *Example:*

```
SDIA 5 = SDIA(4)
```

### *Related Keywords:*

UPDATE

## **SETAIM**

### *Purpose:*

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**

### *Purpose:*

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.

*Example:*

```
SETAIMDATA 5, 0.34
```

*Related Keywords:*

SETAIM

## **SETAPODIZATION**

*Purpose:*

Sets the system apodization type and factor.

*Syntax:*

```
SETAPODIZATION type, factor
```

*Discussion:*

This keyword requires the type to evaluate to an integer expression. The type is 0 for none, 1 for Gaussian and 2 for tangential. The factor sets the apodization factor. See "Apodization Type" on page 86.

*Example:*

```
SETAPODIZATION 1, 0.7
```

## **SETCONFIG**

*Purpose:*

Sets the current configuration for multi-configuration (zoom) systems.

*Syntax:*

```
SETCONFIG (confignumber)
```

*Discussion:*

This keyword requires one numeric expression that must evaluate to an integer between 1 and the maximum number of configurations. The expression is evaluated and then rounded down to an integer to yield the configuration number.

*Example:*

```
SETCONFIG 4
```

*Related Functions:*

CONF, NCON

## **SETDETECTOR**

*Purpose:*

Sets the coherent or incoherent detector data for any pixel on a detector rectangle or detector surface 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 either a detector surface or 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” on page 332. The units for incoherent intensity are source units (see “Source Units” on page 87). The units for coherent amplitude are the square root of source units.

*Example:*

```
SETDETECTOR 1, 5, 12, 0, 1.0056
```

*Related Functions:*

NSDD

## **SETMCOOPERAND**

*Purpose:*

Sets any row or configuration of the Multi-Configuration Editor to any numeric value.

*Syntax:*

```
SETMCOOPERAND 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” on page 471.

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:*

```
SETMCOOPERAND 3, 4, somevalue, 0  
SETMCOOPERAND 1, 0, "THIC", 0
```

*Related Functions:*

MCOP

## **SETNSCPARAMETER**

*Purpose:*

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**

### *Purpose:*

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**

### *Purpose:*

Sets properties of NSC objects.

### *Syntax:*

SETNSCPROPERTY surface, object, code, csg, 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 coating and scatter group 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
0	Sets the object type. The value should be the ASCII name of the object, such as "NSC_SLEN" for the standard lens. The ASCII 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_".
1	Sets the object comment.
2	Sets the reference object number.
3	Sets the "inside of" object number.
4	Sets the object material.
5	Sets the coating name for the specified csg.
6	Sets the profile name for the specified csg.
7	Sets the scatter mode for the specified csg, 0 = none, 1 = Lambertian, 2 = Gaussian, 3 = ABg.
8	Sets the scatter fraction for the specified csg.

Code	Property
9	Sets the number of scatter rays for the specified csg.
10	Sets the Gaussian sigma for the specified csg.
11	Sets the reflect ABg data name for the specified csg.
12	Sets the transmit ABg data name for the specified csg.
13	Sets the "User Defined Aperture" flag. Use 1 for checked, 0 for unchecked.
14	Sets the "User Defined Aperture" filename.
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 diffraction splitting.
85	Sets the parameter value to pass to the DLL, where the csg 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 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	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 csg 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 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.

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**

**Purpose:**

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 Hx-Py, 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. 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**

**Purpose:**

Sets surface tilt and decenter data.

**Syntax:**

```
SETSTDD surf, parameter, value
```

**Discussion:**

This keyword requires 2 numeric expressions that evaluate to integers specifying the surface and tilt/decenter parameter, and a third parameter for the new value. The parameter is used as follows:

0: Before order. 0 for dec/tilt, 1 for tilt/dec.

1-5: Before decenter x, decenter y, tilt x, tilt y, tilt z, respectively.

6: After status. 0 for explicit, 1/2 for pick up/reverse current surface, 3/4 for pick up/reverse current-1 surface, 5/6 for pick up/reverse current -2 surface, etc...

7: After order. 0 for dec/tilt, 1 for tilt/dec.

8-12: After decenter x, decenter y, tilt x, tilt y, tilt z, respectively.

**Example:**

```
SETSTDD 3, 1, before_dx_value
```

**Related Functions:**

STDD

## **SETSURFACEPROPERTY**

**Purpose:**

Sets properties of surfaces.

**Syntax:**

```
SETSURFACEPROPERTY surface, code, value
```

**Discussion:**

This keyword requires 2 numeric expressions that evaluate to integers specifying the surface number, and a code which specifies what property of the surface is being modified. The third 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
0	Sets the surface type. The value should be the ASCII name of the object, such as "STANDARD" for the standard surface. The ASCII 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.
1	Sets the comment.
2	Sets the curvature (not radius) in inverse lens units. Use zero for an infinite radius.
3	Sets the thickness in lens units.
4	Sets the glass.
5	Sets the conic constant.
20	Sets the surface aperture type. See APTP on page page 550 for the aperture type codes.
21, 22	Sets the surface aperture min, max value respectively.
23, 24	Sets the surface aperture decenter x, y value respectively.
25	Sets the User Defined Aperture (UDA) file name.
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: "Reference Radius". Use 0 for best fit, 1 for shorter, 2 for longer, 3 for x, 4 for y, 5 for plane.
80	Sets the scatter code; 0 for none, 1 for Lambertian, 2 for Gaussian, 3 for ABg.
81	Sets the scatter fraction, should be between 0.0 and 1.0.
82	Sets the Gaussian scatter sigma.
83	Sets the ABg name.

Other code values will be added in the future to set other properties of surfaces as requested by users. Some of the code values used are redundant with other ZPL keywords, for example, the command

```
CURV 5 = 23.4
```

is equivalent to

```
SETSURFACEPROPERTY 5, 2, 23.4
```

The function SPRO uses a very similar syntax and identical code values to "get" rather than "set" these same values. Usually, changes to surface properties do not become effective until after the UPDATE keyword is executed.

*Example:*

```
SETSURFACEPROPERTY 1, 4, "BK7"
```

*Related Functions:*

SPRO

*Related Keywords:*

UPDATE

## SETTEXTSIZE

*Purpose:*

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

*Purpose:*

Sets the title of the lens that normally appears on all plots.

*Syntax:*

```
SETTITLE A$
SETTITLE "Here is the lens title"
```

## SETUNITS

*Purpose:*

Sets the current lens units.

*Syntax:*

```
SETUNITS code
```

*Discussion:*

This keyword requires one numeric expression that must evaluate to an integer between 0 and 3. The code is 0 for mm, 1 for cm, 2 for inches, and 3 for meters. This function does not scale or convert the lens data in any way, it only changes how the lens prescription data is interpreted.

*Example:*

```
SETUNITS 0
```

## SETVAR

### *Purpose:*

Changes the state of variables for optimization.

### *Syntax:*

```
SETVAR (surface_expression), VARCODE, (status_expression), (object_number)
or
SETVAR (config), M, (status_expression), (operand_number)
```

### *Discussion:*

The surface\_expression must evaluate to an integer argument between 0 and the maximum number of surfaces, or an error will result. The surface\_expression determines which surface will be made variable. The VARCODE must be one of the following ASCII mnemonics (or a string variable containing the mnemonic):

```
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 the status\_expression evaluates to 0, then the variable status is removed. Otherwise, the value is made variable. If the VARCODE is N<sub>n</sub> or O<sub>n</sub>; the object number must be provided; otherwise, it should be omitted. If the VARCODE 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

### *Purpose:*

Changes the maximum size of the VEC1, VEC2, VEC3, and VEC4 arrays.

### *Syntax:*

```
SETVECSIZE (expression)
```

### *Discussion:*

The expression must evaluate to an integer argument between 1 and 2,000,000. 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

### *Purpose:*

Sets the vignetting factors for the lens.

### *Syntax:*

```
SETVIG
```

### *Discussion:*

See the Chapter "Conventions and Definitions" for a description of vignetting factors.

## SHOWFILE

### *Purpose:*

Displays a text file to the screen using the ZEMAX file viewer.

**Syntax:**

```
SHOWFILE "filename" saveflag  
SHOWFILE NAME$ saveflag
```

**Discussion:**

The filename must be a valid filename within quotes or a string variable containing a valid file name. The file must be an ASCII file (as would be created by OUTPUT and PRINT statements in ZPL) and must be in the current directory. 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 statements. SHOWFILE also closes the file if no CLOSE statement 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

**SOLVETYPE**

**Purpose:**

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_expression), CODE, (arg1), (arg2), (arg3)
```

**Discussion:**

The surf\_expression must evaluate to an integer argument between 0 and the maximum number of surfaces, or an error will result. The CODE must be an ASCII mnemonic as listed in the following table, or the name of a string variable holding the mnemonic. The arg1, arg2, and arg3 expressions evaluate to the first, second, and third solve parameters as specified in "SOLVES" on page 379. For non-sequential pickup solves, the arguments are the first, second, and third lines that follow the "Solve Type" selection on the NSC solve dialog box. Some codes do not use any or all of the arguments, but dummy values for the unused arguments (such as 0) must still be provided and be separated by commas or a syntax error will result.

**CODES FOR SOLVETYPE KEYWORD**

Solve Type	CODE
Curvature: Fixed (turn solve off)	CF
Curvature: Pickup	CP
Thickness: Fixed (turn solve off)	TF
Thickness: Edge Thickness	TE
Thickness: Marginal Ray Height	TM
Thickness: Pickup	TP
Glass: Fixed (turn solve off)	GF
Glass: Model	GM
Glass: Offset	GO
Glass: Pickup	GP

Solve Type	CODE
Conic: Pickup.	KP
Parameter: Parameter 0-16 Pickup.	P0, P1, P2...P16
Non-Sequential Component Pickup X, Y, Z, Tilt-X, Tilt-Y, Tilt-Z, Material. Append an underscore then the object number to the code, for example, NSC_PX_14 will set a pick up solve on object 14.	NSC_PX, NSC_PY, NSC_PZ, NSC_PTX, NSC_PTY, NSC_PTZ, NSC_PMAT (see left)
Non-Sequential Component Parameter Pickup. Append an underscore then the object number, then another underscore and the parameter number to the code, for example, NSC_PP_11_7 will set a pick up solve on object 11, parameter 7.	NSC_PP (see left)

**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
```

**Related Functions:**

SOLV

## **STOPSURF**

**Purpose:**

STOPSURF sets the current stop surface location by number.

**Syntax:**

STOPSURF (surface)

**Discussion:**

This keyword requires one expression which evaluates to an integer between 1 and the number of surfaces minus one. The UPDATE command must be issued before the new data takes effect.

**Example:**

```
STOPSURF n+2
```

**Related Keywords:**

UPDATE

## **SUB**

See GOSUB.

## **SURFTYPE**

**Purpose:**

Changes the surface type.

**Syntax:**

SURFTYPE (surf\_expression), CODE

**Discussion:**

The surf\_expression must evaluate to an integer between 0 and the number of surfaces. The CODE is an integer value which indicates the surface type. The integer codes may be determined at run time by the SCOD function. The SURFTYPE command cannot be used to set a grid sag, grid phase, or user defined surface.

**Example:**

```
SURFTYPE j+1, SCOD("STANDARD")
B$ = "EVENASPH"
SURFTYPE 5, SCOD(B$)
```

## TELECENTRIC

### *Purpose:*

Sets the telecentric object space mode. See “Telecentric Object Space” on page 87.

### *Syntax:*

```
TELECENTRIC = (new_value)
```

### *Discussion:*

If (new value) evaluates to 0, telecentric mode is turned off; otherwise, telecentric mode is turned on.

## TESTPLATEFIT

### *Purpose:*

TESTPLATEFIT calls the test plate fitting routine. See “Test Plate Fitting” on page 197.

### *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 directory; 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

### *Purpose:*

THIC is used to set the thickness of a surface.

### *Syntax:*

```
THIC (surface) = (new_value)
```

### *Discussion:*

This keyword requires two expressions, one to specify the surface number and the other to define the new value. The expression (surface) is evaluated and then rounded down to an integer to yield the surface number. If the surface number is less than zero or greater than the number of surfaces, the command is ignored. Otherwise, the expression (new\_value) is evaluated and assigned.

### *Example:*

```
THIC i+m = THIC(i+n)-1
```

### *Related Keywords:*

UPDATE

## TIMER

### *Purpose:*

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"
```

## UNLOCKWINDOW

*Purpose:*

Unlocks any locked window.

*Syntax:*

```
UNLOCKWINDOW winnum
```

*Discussion:*

See "Graphic windows operations" on page 36.

*Example:*

```
UNLOCKWINDOW 2
```

*Related Keywords:*

LOCKWINDOW

## UPDATE

*Purpose:*

Updates pupil positions, index data, paraxial constants, semi-diameters, maximum field normalization values, and solves. The UPDATE keyword MUST be used before tracing or evaluating a system if the prescription data (such as radii, thicknesses, system apertures, wavelengths) have been altered since the last UPDATE. If the UPDATE command is followed by the keyword "ALL", then all open 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.

*Syntax:*

```
UPDATE
UPDATE ALL
UPDATE n
```

*Example:*

```
THIC 5 = THIC (5)-1
UPDATE
```

## VEC1, VEC2, VEC3, VEC4

*Purpose:*

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 (array_subscript) = (new_value)
VEC2 (array_subscript) = (new_value)
```

*Discussion:*

VEC1..4 are used to store any user data in an array. The array\_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, or an error will be reported. The ZPL functions VEC1..4 can be used to extract the data.



To change the default size to a value different from 1000, 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

*Purpose:*

WAVL and WWGT are used to change the wavelength value and weight.

*Syntax:*

```
WAVL (wavelength number) = (new_value)
WWGT (wavelength) = (new_value)
```

*Discussion:*

This keyword requires two expressions, one to specify the wavelength number and the other to define the new value. The expression (wavelength number) is evaluated and then rounded down to an integer to yield the wavelength number. If the wavelength number is less than 1 or greater than the number of wavelengths, the command is ignored. Otherwise, the expression (new\_value) is evaluated and assigned. The UPDATE command must be issued before the new data takes effect.

*Example:*

```
WAVL 1 = 0.486
WAVL 2 = 0.587
WWGT 1 = 1.0
WWGT 2 = 0.65
```

*Related Functions:*

WAVL, WWGT

*Related keyword:*

NUMWAVE

## XDIFFIA

*Purpose:*

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" on page 528. For a description of the Extended Diffraction Image Analysis feature see "Extended Diffraction Image Analysis" on page 151.

*Syntax:*

```
XDIFFIA outfilename infilename
XDIFFIA out$ in$
```

*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\Beamfiles subdirectory. The infilename must be placed in the \IMAFfiles subdirectory. 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 XDIFFA will use the saved settings. The exceptions are the output file name, which is specified as the first argument after the XDIFFA keyword, and the input source file, which is optionally specified as the second argument after the XDIFFA keyword.

*Example:*

```
XDIFFA "output will be in this file.ZBF" "SOMEIMAFILE.IMA"
```

*Related Functions:*

ZBFCLR, ZBFSHOW, ZBFSUM, ZBFMULT

## ZBFCLR

*Purpose:*

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" on page 528. 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\Beamfiles subdirectory. All ZBF output files will be placed in this same directory.

*Example:*

```
ZBFCLR "some beam file name.ZBF"  
ZBFCLR N$
```

## ZBFMULT

*Purpose:*

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 "ZBFCLR" on page 602 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

*Purpose:*

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, and wavelength (in lens units); the values are placed in vector positions 1 through 11.

See "ZBFCLR" on page 602 for comments that apply to all ZBF related keywords.

*Example:*

```
ZBFPROPERTIES "TEST.ZBF", 1  
rayleighx = VEC1(9)
```

## ZBFRESAMPLE

*Purpose:*

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 resulting data is written back to the same file name.

See "ZBFCLR" on page 602 for comments that apply to all ZBF related keywords.

*Example:*

```
ZBFRESAMPLE "TEST.ZBF", 128, 128, 25.4, 25.4, 0, 0.4
```

## ZBFSHOW

*Purpose:*

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.

See "ZBFCLR" on page 602 for comments that apply to all ZBF related keywords.

*Example:*

```
ZBFSHOW "new beam data.ZBF"  
ZBFSHOW N$
```

## ZBFSUM

*Purpose:*

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 and point spacing in both x and y directions, the second source file listed is first scaled and interpolated as required to match the sampling and point spacing of the first file, then the sum is performed. The outfilename may be the same as one of the source file names; in which case the original file is overwritten.

See "ZBFCLR" on page 602 for comments that apply to all ZBF related keywords.

**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**

**Purpose:**

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 beam 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 resulting data is written back to the same file name.

See "ZBFCLR" on page 602 for comments that apply to all ZBF related keywords.

**Example:**

```
ZBFTILT "TEST.ZBF", 0.0, 0.0, 0.0, 0.01237
```

**Example 1**

Suppose you needed a macro to print out the image plane intercept coordinates of the chief ray at every defined field angle. This is more commonly required than you may think, for example, ZEMAX spot diagrams are all plotted relative to the chief ray coordinates, and so the intercept coordinates are needed to properly offset the data. The same is true for transverse ray aberrations; they are referenced to the chief ray.

The best way to learn ZPL is to study previously written macros. Look closely at the following macro and try to follow the macro logic.

```
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 angle in degrees, 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-angles 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 statements will output the chief ray coordinates on the image plane to the screen. If you like, try typing the macro into a file ending in ZPL (for example, CHIEF.ZPL) and then run it from within ZEMAX

to test it. The coordinates reported should be the same as those listed on the spot diagram when the chief ray is the reference point.

## **Example 2**

Here is an example ZPL macro to 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
wvrms = 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)
    wvrms = wvrms + ( WWGT (w) * rms )
    weightsum = weightsum + WWGT(w)
    FORMAT .4
    PRINT "RMS spot radius for ", WAVL(w),
    FORMAT .6
    PRINT " is ", rms
NEXT
wvrms = wvrms / weightsum
PRINT "Wavelength weighted rms is ", wvrms
FORMAT .2
t = ETIM()
PRINT "Elapsed time was ",t," seconds."
```

Note the use of the trailing commas in the first two PRINT statements. These allow the data printed from the first three PRINT statements to appear on the same line. The intervening FORMAT statements 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.



## **Introduction**



*This feature is only available in the EE edition of ZEMAX.*

---

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” on page 535.*

---

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 "Extensions" menu in the ZEMAX program. The Extensions must be placed in the directory \Extend 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 directory \UDO 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 the chapter "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. This chapter defines the DDE interface to ZEMAX, so that the user may write software programs which take advantage of the ZEMAX ray trace engine.

## **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" 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 communication 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. 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.

## **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. 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 CloseUDOData is:

`CloseUDOData, buffercode`

See also `GetUDOSystem` and `SetUDOItem`.

### **DeleteConfig**

This item inserts deletes an existing configuration in the multi-configuration editor. The syntax is:

`DeleteConfig, config`

See also `InsertConfig`.

### **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, exportdummy, usesolids, raypattern, numrays, wave, field, deletevignetted, dummythick, split, scatter, usepol, config`

For details on this feature and the meaning of all parameters, see "EXPORTCAD" on page 556, and "Export IGES/SAT/STEP Solid" on page 208.

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" on page 68 for details. See also SetAperture.

### 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, numbermcooper"

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"

The parameter type is an integer; either 0, 1, or 2, for angles in degrees, object height, or paraxial image height, respectively. The parameter number is the number of fields currently defined. 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" on page 93.

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`.

### GetMetaFile

This item creates a Windows Metafile of any ZEMAX graphical analysis plot. The syntax is

`GetMetaFile, "metafilename", type, "settingsfilename", flag`

The metafilename must be in quotes and include the full path, name, and extension for the metafile.

The type argument is a 3 character case-sensitive label that indicates the type of analysis to be performed. The 3 letter labels are identical to those used for the button bar in ZEMAX. A list of codes may be found on the "Buttons" tab of the File, Preferences dialog box. The labels are case sensitive. If no label is provided or recognized, a 3D Layout plot 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 metafile graphic, 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 graphic 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 graphic, and not text files, are supported by GetMetaFile.

See also GetTextFile and OpenWindow.

### 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" on page 471. 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 on the General data dialog box.

### 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:

`GetNSCData, 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	Date 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)
3	1 if object uses a user defined aperture file, 0 otherwise. (integer)
4	User defined aperture file name, if any. (string)
5	Reference object number. (integer)

Code	Date returned by GetNSCObjectData
6	Inside of object number. (integer)
7 and up	Reserved for future expansion of this feature.

### 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`.

### 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`.

### 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 type, 2 for int1, 3 for int2, 4-7 for hx-py, 8 for target, 9 for weight, 10 for value, and 11 for percent contribution. The returned string is the numeric value of the requested parameter. 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 directory where ZEMAX is installed, and the path name to the default directory 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:

"nlsPolarized, Ex, Ey, Phax, Phay"

If nlsPolarized 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`.

### 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 number shown on the general data dialog box.

### 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.

### GetRefresh

This data item causes ZEMAX to copy the lens data from the LDE 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" on page 634 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
1, thickness	solvetype, parameter1, parameter2, parameter3
2, glass	solvetype (for solvetype = 0) solvetype, Index, Abbe, Dpgf (for solvetype = 1, model glass) solvetype, pickupsurf (for solvetype = 2, pickup) solvetype (for solvetype = all other values)
3, semi-diameter	solvetype, pickupsurf
4, conic	solvetype, pickupsurf
5-12, parameters 1-8	solvetype, pickupsurf, offset, scalefactor
1001+, extra data values 1+	solvetype, pickupsurf, scalefactor



The solvetype is an integer code, and the parameters have meanings that depend upon the solve type; see the chapter “SOLVES” on page 379 for details. See also SetSolve.

### GetSurfaceData

This item gets surface data. The syntax is:

`GetSurfaceData, surf, code`

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.

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)
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 pick up this surface, 2 for reverse this surface, 3 for pick up 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.

For example, to get the conic data for surface 8, the item name should be "GetSurfaceData,8,6".

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 "GetSurfaceParameter,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, useenvdata, 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 false, that is it is axial, or 1 if the system is not axial), the ray aiming (0 for off and 1 for on), the use environment data flag (0 for no, 1 for yes), the current temperature and pressure, and the global coordinate reference surface number.

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.

### GetTextFile

This item creates an ASCII text 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. The 3 letter labels are identical to those used for the button bar in ZEMAX. A list of codes may be found on the "Buttons" tab of the File, Preferences dialog box. 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 "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, 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 GetMetaFile and 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"

## 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 apodization, GetTraceDirect does not. Both include surface apodizations.

Although GetTraceDirect is easy to program and use, there is a significant disadvantage to using GetTraceDirect: 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.

### 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" on page 79. For information on importing grid surface data, see "Importing grid data" on page 259.

### 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.

### 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.

### LoadFile

Loads a ZEMAX file into the server. 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 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. 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.

### 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" on page 634 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,ClientWindow,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" on page 634 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,ClientWindow,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".

### 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.

### 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 3 letter labels are identical to those used for the button bar in ZEMAX. A list of codes may be found on the "Buttons" tab of the File, Preferences dialog box. The labels are case sensitive. See also GetMetaFile.

## Optimize

Optimize calls the ZEMAX Damped Least Squares optimizer. The syntax is

`Optimize, n`

where n is the number of cycles to run. 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.

## 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 under File, Preferences on the Editors tab, and is described in "Editors" on page 61. 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.

## 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.

## 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\SAMPLES\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.

## 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" on page 68 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" on page 69.

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,.....`; `SetBuffer,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`

or



SetField, n, xf, yf, wgt, vdx, vdy, vcx, vcy, van

If the value for n is zero, then the field type and the total number of fields is set to the new integer values. 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" on page 471.

See also GetMulticon.

### SetNSCObjectData

This item sets the various data for NSC objects. The syntax is:

SetNSCData, surf, object, code, data

Surf and object refer to the surface number and object number. The code is an one of the integer values defined in the table in the discussion for "GetNSCObjectData" on page 613. The returned value is the data item indicated.

### 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.

### 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.

### 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 hx-py, 8 for target, and 9 for weight. The returned string is the same as for GetOperand. To set the operand type, the value can be 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” on page 634 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, datatype, data

Where the values for datatype are the same as documented for the GetSurfaceData item. The return string is the newly set surface data formatted as documented for the GetSurfaceData item. See also GetSurfaceData 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.

### 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.

### SetUDOData

This item is used to pass data 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 detail in the chapter "Optimization". See that chapter for details. The syntax for SetUDOSystem is:

```
SetUDOData, buffercode, data0, data1, data2,...data50
```

where buffercode is an integer passed to the UDO on the command line. When the ZEMAX server receives the SetUDOData item, it places the data values in a buffer, then closes the buffer for further input. This item allows multiple items to be passed at once to the server, however, there is a limit of 255 characters in the input line. To circumvent this limitation (of Windows) use the SetUDOIItem and Close UDOData items. See also GetUDOSystem.

### SetUDOIItem

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 detail in the chapter "Optimization". See that chapter for details. The syntax for SetUDOIItem is

```
SetUDOIItem, 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. Using this item, the server's buffer may be filled one item at a time, which largely circumvents the 255 character limit on item names. However, the ZEMAX server does not know if any more data is coming, and so the buffer remains open. 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:

```
SetUDOIItem, buffercode, 0, value0  
SetUDOIItem, buffercode, 1, value1  
SetUDOIItem, buffercode, 2, value2  
CloseUDOData, buffercode
```

See also GetUDOSystem.

### SetVig

This item sets the vignetting factors. For more information, see "Vignetting factors" on page 96. 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 its 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; /* 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[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; /* 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[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 */

```

If all size of the electric field values are zero; then ZEMAX will use the

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" on page 633.

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. Within zclient.c there is a constant called DDE\_TIMEOUT that has a value of 5000 (the units are milliseconds, so this is 5 seconds). 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. Alternatively, DDE\_TIMEOUT can be set at a higher number, but this is not preferred.

## **How ZEMAX calls the client**

ZEMAX looks in the \Extend directory for Extension executables. Each program in this directory 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 "Extensions" on the main menu bar.

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\IDDE\_DEMO.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 GetSettingsData 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 ASCII 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 (See the File Menu chapter under Environments for details). 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 upper left and lower 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 time (depending upon the User's Environment 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.

### **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 File, Environments dialog. 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 \Extend directory 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 Extensions, then DDE\_DEMO. Note that any .EXE application placed in the \Extend directory will be listed in the ZEMAX Extensions 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 communication, then calls UserFunction. Within UserFunction, two functions provided by ZCLIENT, PostRequestMessage 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.

### **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 and GetMetaFile*

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 directory. For more information, see "GetTextFile" on page 618 and "GetMetaFile" on page 612.





---

**Symbols**

---

$\mu\text{m}$  87

---

**A**

---

- Abbe number
  - Tolerance on 448
- Abbe V number
  - Definition for 485
  - For MILNUM glasses 485
  - Operand 405, 407
- ABCD surfaces 228
- Aberration
  - Longitudinal 160
  - Pupil 110
  - Ray fans 108
  - Wavefront 128
- Aberration coefficients 164, 165, 416
  - Annular Zernike 171
  - Extended Zernike 169
- ABg model 354
  - Catalog of data 203
  - Wavelength scaling of data 204
- Aborting
  - Editing of cell values 36
  - Long computations 40
- Absorbers 347
- Absorption 484, 500
- Active configuration
  - Defined 45
- Active Cursor 37
- Address box
  - Preferences 58
- Afocal systems 266
- Air pressure 491
- Airy disk
  - Defined 111
- Alternate Even surface 228
- Alternate Odd surface 229
- Alternate surface intersection 228, 229
- Analysis units 87
- Anamorphic asphere surface 278
- Angle of incidence 414
- Angular distributions 332
- Angular magnification
  - Defined 45
  - Operand 393
- Angular Spectrum 530
- Angular spectrum 517
- Animation 218
- Annotate
  - Graphics 37
- Annotations
  - On graphic windows 38
- Annular aspheric surface 278
- Annulus object 298
- Anti-nodal planes 45
- Anti-principal planes 45
- Aperture
  - System aperture value 85
- Aperture checking
  - With footprint diagram 160
- Aperture type 68
- Apertures 69
  - Floating 69
  - Removing 206
  - Soft edged 279, 280
  - Surface aperture 53
  - System aperture 53, 85
  - User defined 69
- Aplanatic condition
  - Solving for 381
- Apodization 471
  - Cosine cubed 86
  - Defined 45
  - Gaussian 87, 390
- Apodization factor 471
  - Defined 45
  - Specifying 87
- Apodizations
  - User defined 45, 277
- Appending
  - Lens files 58
- AR
  - Codes 479
- Arms
  - For pupil integration 388
- Array
  - Of diodes 337
- Array variables
  - ZPL macros 596, 600
- Arrays
  - Of lenses 278
- Arrays of objects 215
- Arrow
  - Drawing on plots 38
- Arrow source
  - Size of 96
- Aspect ratio 37, 39
  - Printing 60
- Aspheric object 308, 309, 318
- Aspheric surface object 298, 299, 331
  - Hologram 313

- Aspheric surfaces 243
  - Fast tracing 92
  - High order terms 244, 246
  - Superconic 271
- Asterisk
  - Hyperhemispheric surfaces 52
  - In data cells 61
  - Indicates surface apertures 65
- Astigmatism
  - Calculating 164
  - Operand 393
- Athermal lenses
  - Analyzing and optimizing 496
- Atmospheric refraction 229
- Atmospheric turbulence 258
- Axial color
  - Operand 394
- Axial symmetry 389
- Axicon 270
- Axicon surface object 299

## B

---

- Back focal length
  - Defined 45
- Beam footprint 160
- Beam splitter
  - Defining coatings for 352
- Beamsplitter
  - Polarization modeling 512
- Beep command 552
- Beer's Law 512
- Bending
  - Using fold mirrors 207
- Best fit glass 219
- Best fit sphere 177
- Biconic 226
  - Surface type 229
- Biconic lens object 300
- Biconic object 300, 301
- Bicubic spline 259
- Binary 1 object 302
- Binary 2 object 303
- Binary optics 230, 232, 282, 284
  - Sign conventions for 232
- Biocular analysis 154, 155
- Birefringent materials 235
- Bitmap images 146
- Bitmaps
  - As sources 322
  - Exporting 42, 556, 557
- BMP 146
  - Creating 37

- Boresight error 394
  - Tolerancing 441, 455
- Boundary conditions
  - On (center) thickness 396
  - On curvature 396
- Boundary operands
  - Understanding 421
- BSDF 354
- Bubbles 323
- Bugs
  - Reporting 33
- Bulk absorption 352, 512
- Bulk scattering 349
- Bulk transmission 512
- Button bar
  - Defining 63, 64

## C

---

- CAD files
  - Importing 314
- Candela 88
- Capping
  - GRADIUM profiles 256
- Capturing
  - Screen images 42
- Cardinal planes 45
- Cardinal points 45, 178, 220
- Cell
  - Editing contents 36
- Cell width
  - Changing 60, 61
- Celsius
  - Defining temperature 491
- Center thickness
  - Operand 396, 405, 407
- Centroid 111
  - Determining coordinates of 113
  - Encircled energy 135
  - Optimization 410, 417
- Check for updates 61
- Checking for 61
- Chief ray
  - Definition of 45
  - Normalized coordinates 49
  - Solve 380, 381, 382
- Chief ray solve
  - On coordinate breaks 384
- Chord Tolerance 314
- Chromatic aberration 162
- Circular 69

- Clipboard
  - Copying graphics to 37
  - Copying text to 39
  - Cut, Copy, and Paste 75, 76, 77, 78, 81
  - Selecting metafile format 60
  - Using 42
- Cloning windows 37, 39
- Coating file 91, 202, 501
- Coating groups 352
- Coating multiplier 74, 505, 508
- Coatings
  - Adding to surfaces 203
  - Defined by tables 507
  - Editing 202
  - Loading coating file 203
  - Multi-configuration control of 472
  - Optical 184
  - Properties of 184
  - Tapered 503, 505
- Coherence length 350, 359
- Coherent imaging 149
- COLOR
  - ZPL macros 553
- Color
  - True color PSF 127
- Color of rows 67, 348
- Color slides 322
- Colors
  - Changing default 59
  - Defining 63
  - Of objects 348
  - Of surfaces 67
- Coma
  - Calculating 164
  - Operand 396
- Comments 66
  - Hide/Show menu 75
  - In the merit function using BLNK 394
- Company name
  - On plots 58
- Compensators
  - For tolerancing 450
- Compound parabolic concentrator (CPC) 294
- Cone angle 85
- Cone object 304, 305
- Configuration
  - Changing 97, 475
  - Selecting on windows 37
  - Zoom position 97
- Conic
  - Description of 270
  - Solve for 384
  - Tolerance on 442
- Conjugate
  - Surface type 238, 239
- Conrady
  - Dispersion formula 481
- Consider objects list 351
- Constant
  - Operand 396
- Construction optics
  - Constraining 395
  - For holograms 265
- Contours
  - Setting 131
- Convergence 154, 155
- Convert semi-diameters
  - to apertures 203, 206
- Converting to NSC groups 214
- Coordinate axes
  - Definition of 46
- Coordinate breaks
  - And paraxial rays 92
  - Reversing with "order" 239
  - Solves for 384
  - Surface tilts and decenters 73
  - Surface type 239
- Coordinate reference
  - Drawing 349
- Coordinate system
  - Conventions for 46
- Copy
  - Operand data 76, 77, 79
  - Surface data 75, 81
- Corner object 319, 329
- Cosine lamp 339
- Cosine source 339
- Cosine space 112, 332
- Cosine surface 267
- Coupling efficiency 142, 145, 175, 400, 538, 546
- CPC 305
- CPU's
  - Number of 364, 423, 437
- CR
  - Codes 479
- Creating POB files 217
- Cross sections
  - Point Spread Functions 125
- CSG's defined 352
- Cubic spline
  - Extended surface type 245
- Cubic spline surfaces 240
- Curvature
  - Boundary conditions on 396
  - Operand 396
- Cut
  - Surface data 75

- Cylinder arrays 278
- Cylinder Fresnel surface 240
- Cylinder lens object 315, 317, 327
- Cylinder pipe object 305, 306
- Cylinder volume 396
- Cylinder volume object 306, 307
- Cylindrical lens
  - Paraxial model for 267
- Cylindrical surfaces 229, 272, 273
  - Gratings placed on 248, 273

## D

---

- Date
  - Printing on plots and text 61
- DDE 607
- Debugging
  - Tolerancing 468
- Decenters
  - Tolerancing 445
- Decenters and tilts
  - On surfaces 73
  - With coordinate break surfaces 239
- Decimals
  - Changing default 60, 61
- Default merit function 385
  - Start at row 398
- Default path
  - For output 59
- Defaults, changing 59
- defined 289
- Delano diagrams 161
- Deleting
  - Operands 76, 77, 78, 81
  - Surfaces 65, 74
  - Surfaces in ZPL macros 554, 558
- Density
  - Discussion of 220
  - Of lenses 220
- Detector object 332, 334, 335, 348
- Detector viewer 369
- Detector volume 335
- Dialogs 35
- Diameter
  - Operand 398
- Diameter to thickness ratio
  - Constraining 406, 407
- Diattenuation
  - Of coatings 185
- Diffraction 515
- Diffraction grating 307
  - Multiple orders 350
- Diffraction grating object 307

- Diffraction gratings 241, 268
  - On cylinders 273
  - On toroids 273
- Diffraction Imaginary Transfer Function 115, 116, 117, 120, 121, 122, 125, 126, 128
- Diffraction limited
  - Meaning of 46
- Diffraction Modulation Transfer Function 115, 116, 117, 120, 121, 122, 125, 126, 127, 128
- Diffraction Phase Transfer Function 115, 116, 117, 120, 121, 122, 125, 126, 128
- Diffraction Real Transfer Function 115, 116, 117, 120, 121, 122, 125, 126, 128
- Diffractional optics 258
- Digital Mirror Device (DMD) 317
- Diode 337
- Divergence 154, 155
- Directories
  - Default path for output 59
- Directory names
  - Preferences 59
- Dispersion 162, 163, 178, 179, 198
- Dispersion coefficients 480
- Distortion 156, 158
  - Calculating 164
  - Definition of 156
  - F-Theta 156
  - Grid 158
  - Operand 398
  - Representing 146
- Divergence 154, 155
- DLL 275
- dn/dt 495
- dPgF 479
- Drawing
  - Element 104, 105
- Dual zone 234
- Dynamic Data Exchange (DDE) 607
- Dynamic Link Library 275

## E

---

- Edge margin 93
- Edge Spread Function 125, 137
- Edge thickness
  - Controlling 382
  - Defined 46
  - Determining 219
  - Operand 399, 406, 407, 418
  - X- direction 419
- Editing
  - Spreadsheet values 36

- Editor windows 35
    - Menu 65
    - Preferences 61
  - Editors
    - Preferences 60
  - Effective F/# 139, 140, 399
  - Effective focal length
    - Defined 47
    - Displaying 219
    - Operand 398
    - Single element 219
  - Element drawings 104, 105
    - ISO 10110 105
  - Element power 219
    - Solve 381
  - Ellipse
    - Defining 270
  - Elliptical 69
  - Elliptical object 307
  - Elliptical surface
    - Defining 238, 239
  - Encircled energy 135
  - END
    - ZPL macros 568
  - Ensquared energy 135, 136, 138
  - Entrance pupil diameter
    - Defined 47
    - Operand 399
    - Setting 85
  - Entrance pupil position
    - Defined 47
    - Operand 399
  - Entry ports 291
  - Environment
    - Defining 491
    - Temperature and pressure 491
  - EQUA
    - Operand 399
  - Escape
    - Aborting long computations 40
  - Even Asphere 226
  - Even Asphere lens object 308
  - Even Aspheric 308
  - Examples 32
  - Exclude Substitution 478
  - Exit ports 291
  - Exit pupil
    - Significance of 91
  - Exit pupil diameter
    - Defined 47
  - Exit pupil position
    - Defined 47
    - Operand 399
  - Exponential notation 61
  - Exporting
    - Graphics and text 42
    - IGES and STEP files 209, 556, 609
  - Extended cubic spline
    - Surface type 245
  - Extended polynomial
    - Surface type 247
  - Extended Polynomial Lens object 308
  - Extended Polynomial Surface object 309
  - Extended sources 142, 146
    - Encircled energy computations 137
  - Extensions 223
  - Extra data
    - Defined 47, 225
    - Operand 418
    - Optimization of 426
    - ZPL Macros 560
    - ZPL macros 555
  - Extraordinary rays 235
- 
- ## F
- 
- F/#
    - Capped at 10,000 94
    - Curvature solve 379
    - Displaying 219
    - Effective 140
    - For fields, wavelengths 220
    - Method of computing 94
    - Paraxial working 51
    - Working 55
  - Fabrication support 177
  - Faceted object 325, 326
  - Faceted surfaces 249
  - Far field 517
  - Far field pattern 339
  - Far-field cosine space 332
  - Fast Asphere Trace 92
  - Fiber coupling efficiency 142, 145, 175, 400, 538, 546
  - Fiber optic object 328
  - Field angles
    - Conventions for 47
  - Field curvature 140, 142, 153, 154, 156
    - Calculating 164
    - Operand 400
  - Field curvature plot
    - Meaning of 154, 155, 156
    - Offsets on axis 156, 161
  - Field positions
    - Defining 96
    - ZPL macros 576
  - Field type
    - Multi-configuration control of 472
  - File menu 35, 57
-

- File names
  - On plots 58
- Files
  - Finding names in ZPL macros 558
- Filter string 364
- Filters
  - User defined neutral density 277
- Fitting
  - Test plates 197
- Fletch
  - Size of 60
- Fletch Rays 101
- Flip
  - Lens elements 212
- Float by stop size 85
  - Defined 47
- Floating apertures 69
- Focal length
  - Scaling by 213
- Focus
  - Quick focus tool 213
- Fold mirrors
  - Inserting 207, 215
- Font size
  - Changing 60, 61
  - In text windows 61
  - Setting in ZPL macros 595
  - When printing 62
- Footprint
  - Beam 146
- Footprint diagram 160
- FOR loop
  - ZPL macros 558
- Foucault analysis 129
  - As optimization operand 400
- FR
  - Codes 479
- Fraunhofer diffraction 520
- Fresnel
  - General polynomial form 249
  - Zone plates 286
- Fresnel diffraction 519
- Fresnel lens 309, 310, 534
- Fresnel lens object 309
- Fresnel lenses 326
- Fresnel number 516
- Fresnel object 327
- Fresnel surfaces 240, 245, 249
- Fringes
  - Test wavelength 452
- F-theta
  - DISC operand 397
- F-Theta distortion 155, 156
- Full width half max 138

## G

---

- Gases
  - Thermal modeling 484
- Gaussian Beam
  - Analysis 187
  - Operands 400
  - Paraxial analysis 187
  - Physical optics analysis 515
  - Skew analysis 189
- Gaussian factor
  - Optimization 389
- Gaussian Quadrature 387
- Gaussian scattering 354
- Gaussian source 339
- General data 85
- Geometric MTF 119
  - Vs. Field 121
- Geometry errors 364
- Ghost reflections 48, 215, 362, 371
- Glass
  - Best fit 219
  - Codes 479
  - Constraining 425
  - Cost factors 479
  - Dispersion formulas for 480
  - Entering type 48
  - Excluding from optimization 478
  - Gradient index 250, 349
  - Model 219
  - Optimizing 424, 437
  - Optimizing with RGLA 415
  - Pick up solves 383
  - Removing in multi-configurations 475
  - Substitution 383, 426, 437, 478
  - TCE 479
  - Templates 195
  - Transmittance of 163, 179
  - ZPL macros 548
- Glass catalog 162, 163, 178, 179, 198, 203, 477
  - Adding new glasses to 480, 482
  - Copying or moving 479
  - Creating new 479
  - Managing multiple 477
  - Melt data 483
  - Obsolete glasses 490
  - Selecting 477
  - Transmission data 484
- Global coordinates 93, 174, 612, 613
  - From ZPL macros 543
  - Multi-configuration control of 472
  - Operand 401, 414
  - Reference surface 67, 93, 560

- Global optimization 195, 433, 435
- Glue distance 95, 345
- GOSUB 568
- Gouy shift 189, 522
- Gradient index 250, 251, 252, 253, 257, 258
  - Boundary operands for 429
  - Dispersion model 252
  - Operand 403
  - Vignetting 93
- Gradient Lens Corporation 254
- Graphic frames 60
- Graphics
  - Colors 63
  - Copying to clipboard 37
  - Exporting using Clipboard 42
  - Preferences 59
- Grating mode 268
- Gratings 241, 268
  - Elliptical 242, 243
  - Multiple orders 350
  - Toroidal 248, 273
- Grid phase surface 258
- Grid sag surface 259
- Guard band 526

## H

---

- Hammer optimization 195
- Hardware lock
  - Installing 32
  - Policy regarding 33
- Harvey-Shack scattering 354
- Heads-Up Displays (HUD) 154, 155
- Help
  - Button 40
  - Menu option 35
- Herzberger
  - Dispersion formula 481
- Hexapolar rings
  - Defined 48
- Hidden line drawings 100, 102, 103, 107, 108
- Hiding
  - Edges of surfaces 68
  - Rays on layout plots 68
- Highlight color 60, 63
- Highlighting layouts 60

- Holograms 261, 262
  - Aspheric non-sequential 313
  - General polynomial form 230, 232
  - Multiple orders 350
  - Non-sequential solid 312
  - Optically fabricated 264
  - Optimizing construction optics 265
  - Toroidal 273
  - Used in reflection 261
  - Using Zernike coefficients 282, 284
- Huygens
  - Point Spread Function 126, 127
- Hx
  - Defined 49
- Hy
  - Defined 49
- Hyperbola
  - Defining 270
- Hyperhemispheres
  - Preventing with HHCN 403
- Hyperhemispheric 67
  - Described 52
  - Testing for 403
- Hyperhemispheric lens object 323
- Hyperhemispheric object 301
- Hyperhemispheric surface object 324

## I

---

- Ideal coatings 507
- IGES files
  - Exporting 208, 209, 210, 211
  - Importing 314
- Ignoring objects 348
- Illumination 146
- Image analysis 142, 146, 148, 151
  - Aborting long computations 40
  - As optimization operand 403
  - Editing IMA files 143, 149, 152, 153
  - IMA file format 144
- Image height
  - Field definition 47
- Image plane
  - Coordinates 112
- Image space F/#
  - Operand 404
  - Setting 85
- Images
  - Appearance of 142, 146
- Importance sampling 351
- Imported object 314
- Incoherent imaging 149
- Increment
  - tolerancing 455

- Index data
  - Operand 404
  - Optimizing 424
- Index of refraction 482, 485
  - Determining 219
  - Dispersion plot 162, 178
  - Table of values 220
  - Temperature and pressure effects 491
- Inequality
  - Operand 410
- Initial Phase 350
- Input
  - ZPL macros 572
- Input focus 36
- Inserting
  - Operands 76, 77, 78, 80
  - Surfaces 65, 74
  - Surfaces in ZPL macros 554, 555, 572, 573
- Inside of
  - Placing sources 336
- Installation
  - Of the hardware lock 32
- Integer
  - Rounding to using INTE 539
- Intensity 88
- Interferogram
  - Data for surface deformation 282, 283, 284, 285
  - Drawing 128
- Interferometers
  - Using data from 282, 284
- Internet 61
- Invariant 165
- Irradiance 88
- Irregular
  - Surface type 262
- Irregularity 445, 446, 447
- ISO drawings 105

## J

---

- Jones matrix 511
- Jones matrix object 295
- Jones matrix surface 263
- Jones vector 497
- JPG
  - Creating 37
- Jx, Jy 350

## K

---

- Keyboard
  - Shortcuts 40
  - Use of 40
- Keyboard shortcuts 36, 40
- Kinoforms 230, 232

- Knife edge test 129

## L

---

- Lagrange invariant 165
  - Operand 404
- Lagrange multiplier 390
- Lambertian 354
- Landscape orientation 62
- Last zoom 37
- Lateral color 161
  - Operand 404
- Layout
  - Element drawing 104, 105
- Layouts
  - Scale bar width 63
  - Three-dimensional 100, 102, 103, 107, 108
  - Two-dimensional 99
- Learning ZEMAX 32
- Lens catalog 199
- Lens Data Editor
  - Printing contents of 220
- Lens design
  - References on 31
- Lens element
  - Reversing 212
  - Tilting and decentering 212
- Lens files
  - Deleting 58
  - Inserting 58
  - Loading 57, 58, 574, 575
  - Opening 57
  - Recently used list 64
  - Saving 57, 587
- Lens mounts
  - Thermal expansion of 491
- Lens object 323
- Lens search 199
- Lens units
  - Defined 48
  - Specifying 87
- Lenses
  - Plotting to scale 100
- Lenslet object 315, 317
- Lightpipes 294, 295, 304, 305, 320
- Line Spread Function 125, 137
- Linear interpolation 259
- Liquids
  - Thermal modeling 484
- Load and Save buttons 40
- Loading
  - Lens files 58
- Lock window 37, 39, 575



- Long computations
  - Aborting from 40
- Longitudinal aberration 139, 160, 161
  - Operand 405
- Longitudinal aberrations
  - Calculating 164
- Longitudinal color
  - Calculating 164
- Lost energy 364, 365
- LPTD
  - Operand 429
- Ltk 479
- Luneberg lens 280

## M

---

- M Squared 187
- Macros
  - ZPL macros 535
- Magnification
  - Displaying 219
  - Operand 393, 412
- Main window 35
- Marginal ray
  - Definition of 48
  - Normalized coordinates 49
  - Solve 380, 381
- Margins 93
  - Setting for printing 62
- Mass
  - Of lenses 220, 417
- Maximum aspheric deviation 177
- Melt data
  - fitting 483
- Melt Frequency 479
- Merit function
  - Default 385
  - Defining 385
  - Formula for 391
  - Gaussian apodization of 390
  - Ignoring operands in 399
  - Loading and saving 76
  - Loading from ZPL 575, 587
  - Modifying 390
  - Percent contribution of components 420
  - Printing 195, 196, 197
  - ZPL macros 570, 577
- Metafile
  - Creating 37
  - Windows clipboard format 60
- MicroElectroMechanical System (MEMS) 295
- Micrometers 87
- Microns 87
- MIL number glasses 485

- Mirrors
  - Conventions for 53
  - Deleting fold 207, 215
  - Inserting fold 207
  - Specifying 48, 66
- Model glass 219, 383, 424, 485
- Modulation Transfer Function 115, 116, 117, 120, 121, 122, 125, 126, 128
- Monte Carlo ray tracing 363
- Mouse
  - Shortcuts 37, 39

- MTF
  - Aborting from long computation 40
  - Calculating in macros 562
  - Geometric approximation 119
  - Operand 406
  - Optimization of 421
  - Visualizing 142, 146
  - Vs. Field 116, 117, 121
  - Vs. Focus 115
- Multi-configuration 471
  - Operand values 405
  - Optimization and 475
  - Tolerancing 467
- Multi-mode fiber coupling 142, 145
- Multiple CPU's
  - Parallel processing 364, 423, 437
- Multiple orders 350
- Multi-threading 364, 423, 437

## N

---

- Narcissus
  - YNI contribution 419
- Near field 517
- Nesting object limits 345
- Nesting sources 336
- Neural networks 435
- Neutral density filters
  - With user defined surfaces 277
- Nodal points 45, 220
- Non-paraxial system
  - Defined 49
- Non-sequential 293, 336
  - Converting surfaces to 214
  - Defined 49
  - Multi-configuration operands for 473
  - Object definitions 293, 336
  - Optimization of systems 427
  - Optimization operands for 408, 410
  - Ray tracing 289
  - Ray tracing limits 94
  - Restricting 94
- Non-sequential program mode 57

- Normal distribution 460
- Normalized coordinates 390, 579, 583
  - Defined 49
- Note files
  - Editing 104
- Notes 88
- NSC 289
  - With ports 290
  - without ports 290
- NSG America, Inc. 257
- Null object 318
- NURBS surfaces 269, 274

## O

---

- Object
  - User defined 329
- Object plane
  - Tilting 272
- Object space N.A.
  - Setting 85
- Object space numerical aperture
  - Defined 50
  - Operand 410
- Objects 293, 336
  - Arrays of 215
  - Ignoring 348
  - Inserting and deleting 81
  - Overlapping 345
  - Replicating 215
- Objects of revolution 325
- Obscuration type 68
- Obscurations 68, 73
  - Circular 69
  - Elliptical 69
  - Rectangular 69
  - User defined 69
  - Vignetting from 140
- Obsolete
  - glasses 490
- Odd Asphere 227
- Odd Asphere lens object 318
- Offense against the sine condition 411
- Ogive surface 281
- Opacity
  - Of objects 348
  - Of surfaces 67
- OPD
  - Optical path differences 410
- OpenGL
  - Shaded model graphics 103, 108
- Opening
  - Lens files 57

- Operand data
  - Cut, Copy, and Paste 76, 77
- Operands
  - Defining complex 423
  - User defined 429
- Optical design
  - References on 31
- Optical invariant
  - Lagrange invariant 404
- Optical path differences
  - Operand 410
  - Referencing 91
  - Solves 382
- Optically fabricated hologram 264
- Optimization 195, 385
  - Global 195, 435
  - Global minimum 433
  - Global search 435
  - Hammer 195, 435
  - Merit functions for 385
  - Of glass 424
  - Operands 390, 423
  - Requirements 385
  - Selecting variables for 385
  - Suggestions for use 433
  - With external programs 431
  - With multi-configurations 475
  - ZPL macros 429
- Optimization operands
  - Plot of 180
- Orders
  - Splitting rays by 350
- Orientation
  - Graphics 62
- Orientation indicator 101
  - Size of 60
- Overlapping objects 345
- Overlaying graphics 37, 457

## P

---

- Pan and zoom 39
- Parabasal
  - Discussion of 50
- Parabola
  - Defining 270
- Parallel processing
  - Multiple CPU's 364, 423, 437
- Parameter data
  - Defined 50, 225
  - Tolerancing 448
- Parameter solve
  - On coordinate breaks 384

- Paraxial 227
  - Definition of 50
  - Working F/# 51
- Paraxial focus 381
- Paraxial image height
  - Defined 51
- Paraxial lenses 266, 267
- Paraxial magnification
  - Defined 51
- Paraxial ray data
  - Operands 411
- Paraxial rays
  - And coordinate breaks 92
  - Tracing 174
- Paraxial surfaces 266, 267
- Paraxial XY 227
- Paste
  - Operand data 76, 77, 79
  - Surface data 75, 76, 77, 79, 80, 81
- Pasting
  - Graphics and text 42
- Paths
  - For data files 59
- PAUSE
  - ZPL macros 579
- Peak to valley
  - Wavefront error 165
- Pen
  - Changing width of 62
- Pen color
  - ZPL macros 553
- Pen colors 63
- Pen width 60, 62
- Periodic surface 267
- Petzval
  - Operand 411
- Phase
  - Point Spread Function 122
- Phase of a surface 131
- Photometric units 88
- Physical optics 515
- Pick up solves 379
- Pilot Beam 529
- Pilot beam 521
- Pinhole 529
- Pinhole aperture 534
- Placing sources 336
- Plot width 62
- Plots
  - Changing pen width 62
- Plotting
  - All zoom positions 102
  - Optimization operands 180
- POB files
  - Creating 217
- POB format 319
- Point Spread Function
  - Cross sections 125, 127
  - FFT 122, 126
  - For NSC rays 333
  - Huygens 122, 126
- Point spread function 122, 126, 127
- Polarization 525
  - Analysis 497
  - Defining global state 90
  - In non-sequential ray tracing 352
  - NSC sources 350
  - Partial 350
  - Phase of detectors 333
  - Ray trace 182, 190, 193
  - Reflectivity, Transmission, Absorption 184
  - ZPL macros 579
- Polygon object 319
- Polygon objects
  - Defining 373
- Polynomial
  - Extended surface type 247
- Polynomial surfaces 268
- Pop-up screens
  - Configuring 59
- Portrait orientation 62
- Ports in non-sequential tracing 290
- Power
  - Function 542
  - Maintaining with glass change 66
  - Operand 413
  - Solve 381
  - Watts 289
- PR
  - Codes 479
- Preferences 58, 59
  - Address box 58, 59, 60
  - Directory names 59
  - Editors 61
  - Printing 62
- Pre-Propagation 350
- Prescription Data 220
- Pressure
  - Of each surface 495
  - Setting 90
- Pressure effects 491
- Primary wavelength 96
  - Defined 51
  - Multi-configuration 473
- Principal planes 45, 220

## Printing 36

- ASCII from ZPL macros 581
- Changing pen widths 62
- Coating definitions 203
- Contents of a window 219
- Font size 62
- Graphic windows 37
- Landscape mode 63
- Merit function 195
- Orientation of 62
- Portrait mode 63
- Preferences 62
- Prescription data 220
- Spreadsheet data 220
- Surface data 219
- System data 219
- Text margins 63
- Text windows 39, 219
- Tolerance data 196, 197
- Using ZPL macros 580
- Windows in ZPL macros 581

## Prisms 319

## Probing rays 524

## Profiles

- Scattering 91, 349

## Program mode 57

## Propagation report

- For POP analysis 191, 533

## ProSource 341

## Pupil aberration

- Controlling 88

## Pupil positions

- Displaying 219

## Pupil shift 89

## Px

- Defined 49

## Py

- Defined 49

## Q

---

## QSUM

- Operand 414

## Quadratic sum

- Operand 414

## Quality factor 187

## Quick focus 213

## Quitting ZEMAX 64

## R

---

## Radiant Imaging 341

## Radii

- Entering 66

## Radius

- Curvature 396

## Radius of curvature

- Conventions for 51

## Random number generator

- Seeding 452

## Random Polarization 350

## Ray aiming 67, 88

- While tolerancing 457

- ZPL macros 588, 589

## Ray database files 367

## Ray fans 108, 110

## Ray splitting 289, 362

## Ray tracing 174

## Rayleigh range 188, 520, 521, 522, 527

## Raylist format 102

## Raylists 102

## Real propagation 51

## Real ray

- Tracing 174

## Recovering lost files 82

## Rectangular 69

## Rectangular object 320

## Rectangular pipe object 320

## Rectangular volume grating object 322

## Rectangular volume object 322

## Redo 82

## Reference Radius 530

## Reference sphere 91

## References 31

## Referencing OPD

- Described 91

## Reflections

- Ghost 215

## Reflectivity

- Of coatings 184

## Reflectors

- Specifying 48, 66

## Refractive index

- Index of refraction 219

## Relative Cost 479

## Relative cost of glasses 479

## Relative illumination 139, 415

## Remote Desktop 61

## Replicating objects 215

## Report graphics 220

## Resolution

- Estimating 142, 146

## Resolving

- Two field points 113

## Retardance

- Of coatings 186

## RETURN

- ZPL macros 568

- Reverse Rays 350
- Reversing
  - Lens elements 212
- RGLA
  - Operand 425
- Rings 387
- RMS
  - Spot size 112, 132, 133, 134, 385
  - Wavefront error 132, 133, 134, 165, 385
  - X and y spot size 112

- Robust
  - Ray aiming mode 89
- Roofs 321
- Rotating
  - 3D layout plots 101
- Rotation matrix 93, 220, 344
- Row color 67, 348

## S

---

- Sag
  - Constraining 416
  - Operand 416
  - ZPL macros 544
- Sag of a surface 130
- Sag table 177
- Sagittal
  - Definition of 51
- Sampling
  - Hexapolar rings 48
- SAT files
  - Exporting 209
  - Importing 314
- Save and Load buttons 40
- Saving
  - Lens files 57
- Scale bar
  - On layout plots 99
  - Setting accurately 63
- Scale factor 100
- Scale lens 213
- Scaling
  - On printers 63
- Scatter groups 352
- Scatter Towards list 349, 351

- Scattering 352
  - ABg 354
  - Catalog of data 203
  - Gaussian 354
  - Harvey-Shack 354
  - Lambertian 354
  - On sequential surfaces 72
  - On spot diagrams 112, 122
  - Towards an object 351
  - User defined 356
  - Viewer 204
  - ZPL macros 588
- Scattering profile file 91
- Scattering profiles 349
- Schott
  - Constants of dispersion formula 480, 482
- Screen capture 42
- Scripted tolerancing 463
- Searching
  - For stock lenses 199
- Seidel coefficients 164, 165, 169, 171
- SELFOC lenses 257, 258
- Sellmeier
  - Dispersion formula 480
- Semi diameter margin 93
- Semi-diameter
  - Accuracy of 92
- Semi-diameters
  - Accuracy of 52
  - Converting to apertures 206
  - Defined 52
  - Entering 66
- Sequential program mode 57
- Sequential ray tracing 289
  - Defined 52
- Session files 57
- Set to default 385
- Set vignetting
  - On field dialog box 96
  - ZPL macros 596
- Variables 74
- Settings
  - Saving with the lens 40
- SETVECSIZE
  - ZPL macros 595, 596
- Shaded model 103, 108
- Shear distance 258
- Shortcuts
  - Keyboard 35, 36
  - Summary of 40
- Sign conventions
  - For binary optics 232
  - For radius of curvature 51
- Simulated annealing 435

- Single mode fiber coupling 175
- Slide object 322
- Slider control 218
- Soft apertures 279
- Solid modeling 100, 102, 103, 107, 108
- Solves 379
  - Aplanatic 381, 383
  - Center of curvature 383
  - Chief ray angle 380
  - Chief ray height 382
  - Chief ray normal 381
  - Compensator 382
  - Concentric 381, 383
  - Edge thickness 382
  - Marginal ray angle 380
  - Marginal ray height 381
  - Marginal ray normal 381
  - Optical path difference 382
  - Pick up 380, 382, 383, 384, 476
  - Position 382
  - Setting and removing 74
  - Suggestions for use 384
  - Thermal Pick Up 493
  - Turning off 379
- Sorting
  - Tolerance data 79
- Source arrow
  - Size of 96
- Source bitmap 322
- Source diode 337
- Source object
  - Ellipse 339, 343
  - Filament 340
  - File 340
  - Gaussian 342
  - Point 342
  - Rectangle 343
  - Tube 343
  - Volume cylinder 343
  - Volume ellipse 344
  - Volume rectangle 344
- Source units 87
- Sources 336
  - Extended 142, 146
  - User defined 340
- Specular ray 352
- Spherical aberration
  - Calculating 164
  - Operand 416
- Spherical object 323
- Spherochromatism
  - Operand 416
- Spider
  - Spider aperture 69
- Spline
  - Bicubic using grid surface 259
  - Linear interpolation using grid surface 259
- Spline surfaces 240, 269, 274
- Spot diagrams 111, 113
  - Configuration matrix 113
  - Full field 113
  - Matrix 113
- Spot size 132, 133, 134
- Spreadsheet
  - Printing Lens Data Editor contents 220
- Spreadsheet editor
  - Cursor controls for 65
  - Shortcuts 65
- Square root
  - Operand 404, 405, 416
- Square wave MTF
  - Operand 406
- SR
  - Codes 479
- Standard deviation
  - Wavefront error 166
- Standard lens object 323
- Standard surface object 324
- Status bar
  - Modifying 64
- STEP files
  - Importing 314
- STL format 324
- STL Object 324
- STL objects
  - Defining 376
- Stock lens catalog 199
- Stock lenses
  - Using search to find 199
- Stop surface
  - Defining 67
  - Zooming in multi-configurations 474
  - ZPL macros 598
- STP files
  - Exporting 209
  - Importing 314
- Straddling objects 345
- Strehl ratio 46, 52, 126, 127, 132, 133, 134, 167
  - ZPL macros 567
- String length
  - ZPL macros 544
- String operators 549
- String variables
  - ZPL macros 547
- Strings
  - Comparing 544, 545
- Subdirectories 59

- Subroutines
  - ZPL macros 568
- Substitute
  - Glass 383
- Substitution
  - Templates for 195
- Superconic
  - Surface type 271
- Supergaussian 337
- Surface apertures
  - Defined 52
  - Multi-configuration values 471
  - Specifying 68, 73
- Surface data
  - Cut, Copy, and Paste 75, 81
- Surface numbering 65
- Surface Phase 131
- Surface power 219
- Surface Sag 130
- Surface tilts and decenters 73
- Surface types 225
  - ABCD 228
  - Alternate intersection 228, 229
  - Conic 67, 225
  - Coordinate breaks 239
  - Cubic spline 240
  - Diffraction gratings 241, 268
  - Fresnel 249
  - Holograms 261
  - NURBS 269, 274
  - Plane 67, 225
  - Selecting 67, 225
- Surfaces
  - Deleting 65
  - Inserting 65
  - Miscellaneous data 219
  - Reversing order of 212
- System aperture
  - Defined 53
  - Multi-configuration value 471
  - Specifying 85
- System data
  - ZPL Macros 560, 563
- System menu 85

## T

---

- Table coatings 507
- Tabulated Fresnel object 327
- Tabulated object 325, 326
- Talbot imaging 534
- Tangential
  - Definition of 51
- Tapered coatings 503, 505

- TCE 479
  - Adding data 495
  - Data defined 493
  - In glass catalogs 479
- Technical support 33
- Telecentric mode 87
- Temperature
  - Of each surface 495
  - Setting 90
- Temperature effects 491
- Templates for glass substitution 195
- Test plate fit files 198
- Test plate fitting 197
- Test wavelength 452
  - Tolerancing 450
- Text
  - Copying to clipboard 39
- Text data
  - Exporting 564
- Text windows
  - Extracting data from 538
- Thermal effects 491
  - Auto setup 78
- Thermal modeling
  - Of gases and liquids 484
- Thickness
  - Boundary conditions on 396
  - Conventions for 53
  - Operand 417, 418
  - Solves on 381
- Thicknesses
  - Entering 66
- Thin films 184
  - Coatings 184
- Thin lens
  - Cylindrical 267
  - Surface type 266
- Through focus
  - Geometric MTF 120
- Tilted surface 262, 272
- Tilts
  - Tolerancing 445
- Tilts and decenters
  - On surfaces 73
  - With coordinate break surfaces 239
- Time
  - Printing on plots and text 61
- TIR 292, 293
  - Frustrated 512
  - Total Internal Reflection 53
- Tolerance data
  - Cut, Copy, and Paste 78
  - Loading and saving 79

- Tolerance sensitivity
  - Optimizing for 426
- Tolerancing 196, 441
  - Debugging 468
  - Defining compensators for 453
  - Increment 455
  - Irregularity 445
  - Using scripts 463
- Toolbar
  - Defining 63, 64
- Tools menu 195, 223, 289
- Top Hat 527
- Toroidal 228
- Toroidal faceted object 326
- Toroidal Grating 228
- Toroidal lens
  - Paraxial model for 267
- Toroidal lens object 301, 327
- Toroidal surfaces 229, 272, 273
  - Gratings placed on 248, 273
  - Thermal effects 494
- Torus surface 321, 328
- Torus volume 321, 328
- Total Indicator Runout
  - Tolerancing 445
- Total integrated scatter 204
- Total Internal Reflection
  - ZPL Macros 583, 584
- Total track
  - Defined 53
  - Operand 417
- Transfer Functions 115, 116, 117, 120, 121, 122, 125, 126, 127, 128
- Transmission 183, 484
  - Internal 163, 179
  - Of coatings 185
- Transmission data
  - Glass catalogs 484
- Transparent objects
  - Opacity of 348
- Transparent surfaces
  - Opacity of 67
- Transverse aberrations
  - Calculating 164
- Transverse color
  - Calculating 164
- Triangle object 329
- Triangular corner object 329
- True color
  - PSF 127
- Tutorial 32

## U

---

- UDA Scale 69, 348
- Undo 82
  - Disabling 61
- Uniform distribution 460
- Units 87
  - Determining in ZPL macros 545
  - Setting lens units 87
  - Source, irradiance, and flux 87
  - ZPL macros 595
- Universal plot 180
- Unlock window 37, 39, 600
- Unpolarized 90
- Unzoom 37
- Update
  - Described 85
- Update all
  - Described 85
- Updates 61
- Use polarization
  - Polarization state 90
- User defined
  - Apertures and obscurations 69, 352
  - Optimization operands 431
  - Surface 275
  - Surface apodization 277
- User defined object 329
- User defined sources 340
- User defined surface
  - Diffraction optics 277
  - DLL 67
  - Gradient index 277
  - Lenslet arrays 277
  - Refractors and reflectors 276
  - Using linked code 275
  - Using look up table 258, 259

## V

---

- Vacuum
  - Index of refraction 491
  - Modeling 492
- Variable line space gratings 282
- Variables
  - Defining within ZPL macros 596
  - Removing 195
  - Setting and removing 74
  - ZPL macros 586
- Variance
  - Wavefront error 165
- Vector variables
  - Setting the size of 596



- Vignetting 68, 73, 109, 140
  - ZPL macros 583, 584, 596
- Vignetting factors
  - Adjusting during optimization 416
  - Defined 53
  - Field curvature and distortion 158
  - Multi-configuration values 472
  - Removing 206
  - Setting 96
- Virtual propagation 55
- Volume
  - Of an element 219
  - Operand 418
- Volume detector 335
- Volume sources 336
- Volumes
  - Discussion of 220
- Volumetric scattering 349
- Voxels 314, 335

## W

---

- Watts 289
- Wavefront aberrations 164
- Wavefront error 132, 133, 134
  - Peak to valley 165
  - RMS 165
- Wavefront propagation 515
- Wavelengths
  - Conventions for 55, 491
  - Weights for 96
  - ZPL macros 576
- Weight
  - Configuration 472
  - Of lenses 220
- Wide angle apertures
  - In object space 85
- Window size
  - Changing default 59, 60
- Windows
  - Editor properties 61
  - Graphic 35
  - Resizing and moving 35
  - Shortcuts 40
  - Text 35
  - Types of 35
- Wire frame model 103
- Working F/# 85
  - Defined 55
  - Method of computing 94
  - Operand 418
  - Paraxial 51

## Y

---

- YNU tracing 174
- Y-Ybar diagrams 161

## Z

---

- ZBF beam file
  - Decentering 603
  - Properties of 603
  - Resampling 603
- ZBF beam files
  - Macro commands related to 602
- ZBF file format 528
- ZBF files
  - Macro commands for 572, 603
- ZEMAX Beam File 528
- ZEMAX Programming Language
  - ZPL macros 535
- Zernike
  - Deformed surfaces 282, 283, 284, 285
- Zernike coefficients
  - Calculation of 169, 171
  - Fringe 167
  - Optimization of 420
  - Standard 169
- Zernike phase surface
  - Optimizing coefficients 283
- Zernike surface 331
  - Optimizing coefficients 244, 283
- Zone plates 286
- Zoom and pan 39
- Zoom in, out 37
- Zoom lenses
  - Maintaining length of 382
  - Maximum semi-diameter solve 384
  - Multi-configuration feature 471
  - Operand 420
  - Phantom elements 475
- Zoom positions
  - Changing 97
  - On plots 58
  - Plotting all 102
  - Showing number on plots 58, 59
- ZPL macros 223, 224, 535
  - Editing 535
  - Examples 604
  - Optimization of 429
  - Printing 535
  - Status message 535
  - String variables 547
- ZRD file format 367