OpTaliX®

Software for Optical Design and Thin Films

Tutorial

Version 12.10

Optenso™
Optical Engineering Software

support@optenso.com
https://www.optenso.com/

Fritz Blechinger (English version)
Yasunori Murata (Japanese version)

July 2024

Copyright

Trademarks

Disclaimer

Optεnso® reserves the right to modify its software and publications with no obligation to notify any person or organization of such modifications. The software is provided as is without any guarantees or warranty. Although Optεnso® has attempted to find and correct any bugs in the package, in no event shall Optεnso® be liable for any loss of profit or any commercial damage, included but not limited to special, consequential or other damages.

Optenso, Optical Engineering Software
Herbstweg 9
86859 Igling
Germany
E-mail : info@optenso.com
Web : https://www.optenso.com/

This page is blank.

Contents

This page is blank.

1 Installation

OpTaliX is installed by a single install program called `optalix_pro_setup_XXXX.exe`, where XXXX represents a 4-digit version number. The installation file is provided on a CD-ROM or is available by download from https://www.optenso.com/download/download.html. In the Explorer double click on this file and follow the instructions.

1.1 Software Requirements

A 64-bit Microsoft Windows operating system is required to run *OpTaliX* . Supported versions are Windows XP, Windows 7, Windows Vista, Windows 10 + 11.

1.2 Hardware Requirements

minimum:	recommended:
Intel Core I-series (I3) or AMD Ryzen 3 8 GBbyte RAM 200 Mbyte available on hard disc 1024 x 768 pixel monitor resolution	Intel Core I-series (I5, I7) or AMD Ryzen 5 or 7 16 GByte RAM 400 Mbyte available on hard disc 1920 x 1080 pixel monitor resolution or higher

This page is blank.

2

Starting and Exiting OpTaliX

OpTaliX can only be started from within Microsoft Windows. Within Windows, *OpTaliX* can be run by clicking on the *OpTaliX* menu item in the Program Group, double clicking on the *OpTaliX* desktop shortcut icon, double clicking on a lens file in Windows Explorer, or it can be run from a DOS prompt within a DOS window.

2.1 Starting OpTaliX from the Program Group

To start *OpTaliX* in Windows XP/Win7/Win10/Win11, click the **Start** button, click **Programs**, click the *OpTaliX* program group, and then click the *OpTaliX* menu item, as shown in Figure 2.1.

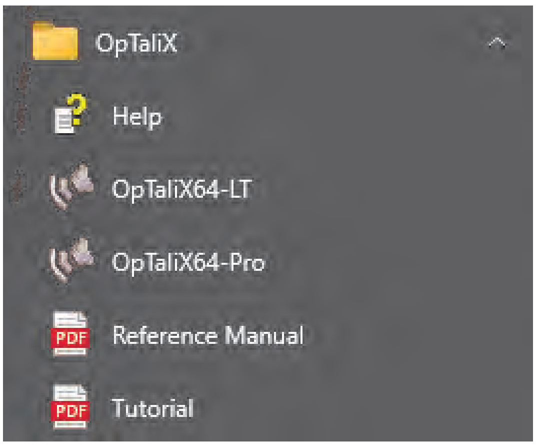


Figure 2.1: *OpTaliX* program group menu.

The *OpTaliX* program group also includes menu items for HTML-Help, Reference Manual, Tutorial and uninstalling *OpTaliX* .

2.2 Starting OpTaliX from Windows Explorer

The *OpTaliX* file format has been registered in Windows during program installation. This allows you to launch *OpTaliX* with a specific lens, by double clicking on the file (extension .otx) in Windows Explorer.

2.3 Starting OpTaliX from a DOS Window

Open a DOS Window by clicking on the MS-DOS prompt menu item in the Program Group accessed by using **Start − > Programs**. From the DOS prompt on a 64-bit operating system, start *OpTaliX* by typing

```
C:> c:\Program Files\optalix-pro\optalix64p mylens.otx
```

respectively, if you have a network license, enter

```
C:> c:\Program Files\optalix-pro\optalix64pn mylens.otx
```

If *OpTaliX* was installed in a different directory than `c:\Program Files\optalix-pro`, the path to the *OpTaliX* executable must be modified accordingly. Specification of an *OpTaliX* lens file (mylens.otx) is optional. If omitted, *OpTaliX* starts with the recently used lens (i.e. the optical design which was in use during the last session). If specified, *OpTaliX* is launched and ”mylens.otx” is automatically loaded.

2.4 Normal Exit from OpTaliX

- From the File menu, select Exit or click on the close window button in the upper right corner of the *OpTaliX* main window.
- In the command line, type `EXI` or `QUIT` and press Return.

After you exit *OpTaliX* , you are immediately returned to the operating system.

2.5 Forced Exit from OpTaliX

Normally an exit request invokes a dialog box asking to confirm exit. Immediate exit by bypassing the confirmation dialog box is accomplished from the command line or from a macro by

`EXI Y`

or

`EXI Yes`

The program is then terminated immediately.

3

First Steps

After launch of *OpTaliX* , the initial screen shows the main window, a text window and a history window as shown in Fig. 3.1.

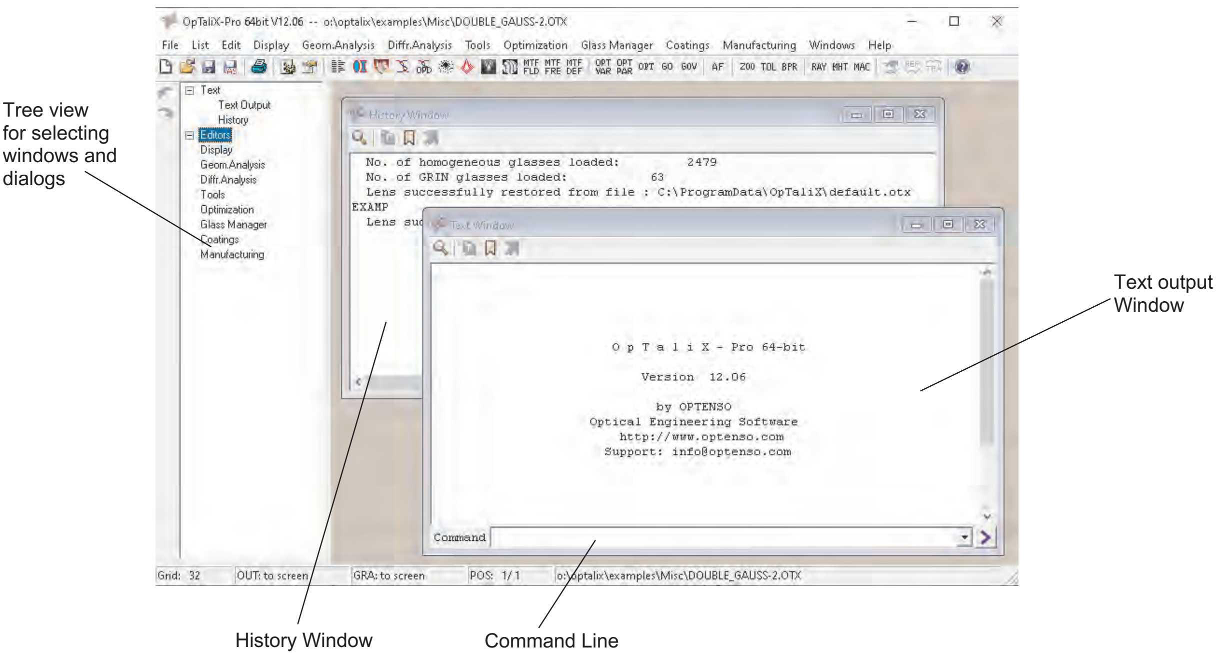


Figure 3.1: *OpTaliX* main window. Note that size and position of the windows may differ from this view, depending on your personal settings.

To the left of the main window is a hierarchical tree view which allows navigating through analysis windows and dialogs. It also helps to easier access windows that are hidden by other windows.

In addition, a text window and a command history window are always opened:

Text Window: The text window outputs all numerical analysis results (such as ray trace data, surface listing, etc.) and information needed during the design process.

History Window: The history window contains, as the name implies, the history of all commands entered, either by user input at the command line or internally generated by the program.

Both windows, the text output window and the command history window are always visible and cannot be closed.

The program can be controlled in two ways, either by commands and macro sequences or from the graphical interface by clicking menus and buttons. Both methods of user interaction work seamlessly together and it is also possible to mix point-and-click operations with the mouse with commands entered in the command line. For almost all commands there exists a selectable menu item in the main menu.

The windows interface provides two command lines, which are fully equivalent in functionality. They are found in the foot of the text window (with a label `Command`, see Fig. 3.1) and a floating command line that is accessible throughout the whole desctop.

The syntax of the command line is universal throughout the program, since it is used for program control, for definition of optimization constraints and also in the macro language. Basically, if a menu item is selected, a command is internally generated, which translates the user response. In the course of this introductory manual we will have a closer look to the commands and its options as the tutorial proceeds.

It is recommended to leave the text window visible as all calculation results, error messages, warnings, etc., are displayed in this window. Both windows, the text window and the history window are always visible and cannot be closed.

Once the program has been completely started, a ''default'' system is always loaded. This is the optical system, which was in memory during the last (previous) session. Also, when the program is terminated, the current system is automatically stored as the ''default'' system. The idea of a default rather than a blank design at program start is to preserve design data between sessions.

3.1 Main Window

The main window is the central focus point and contains the menu bar and the tool bar.

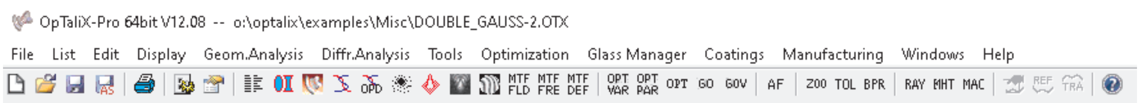


Figure 3.2: Main window, containing menu bar and tool bar.

3.1.1 Menu Bar

The menu bar as shown in Fig. 3.2 displays the top-level items for the *OpTaliX* menu system. Each item contains a pull-down menu with options. Options not available are greyed.

File	Options that restore (open) and save files. It contains the import/export filters for Code V, Oslo, Zemax, Atmos and Modas and setting of preferences. A file-history list for quick access to recently loaded design is also found in this menu.
List	Options for various list outputs such as lens prescription, glasses, coatings, optimization data, etc.
Edit	Options that edit lens surfaces, insert, delete or reverse surfaces, configuration data, zoom data, Zernike surface data, etc.
Display	Options that display data such as lens layout plots or global coordinates.
Geom.Analysis	Options for geometrical performance analysis of the system such as transverse ray aberrations (Fan), spot diagrams, astigmatism, distortion, third-order, polarization, transmission, etc.
Diffr.Analysis	Options for diffraction based calculations such as MTF, PSF, wavefront aberration, edge trace, line-spread function, encircled energy, interferogram analysis and coupling efficiency.
Tools	Advanced or special purpose options such as autofocus, converting MIL-glasses to nearest catalogue glasses, analytical setup of simple optical systems and telescopes, user defined graphics and macros.
Optimization	Options that set up and run optimization.
Glass Manager	Options for viewing optical properties of pre-stored glasses such as the classical $n - \nu$ diagram, partial dispersion diagrams, gradium profile, etc.
Coatings	Options that allow editing and optimizing thin-film coatings including plots for reflection, transmission and phase properties.
Manufacturing	Options that support manufacturing, e.g. test-plate fitting and aspherization plots.
Help	Invokes the help system.

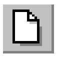
3.1.2 Keyboard Shortcuts (Accelerators)

In addition to selecting menu items using a mouse they can also be accessed via the keyboard. Keyboard shortcuts, known as *accelerators*, provide direct access to specific main menu options. For example, Ctrl+S immediately saves an optical design without the need to select the corresponding menu item by the mouse. To enter Ctrl+S, hold the ''Ctrl'' key pressed and then enter ''S'' on the keyboard.

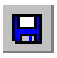
Note that accelerators will not work if the command line, a field in a dialog or a graphics window has focus!

3.1.3 Tool Bar

The tool bar provides quick access to perform often used tasks. A concise online description is shown for each icon if the mouse cursor is held over the icon.

 Makes a new system. Overwrites data of the previous system.

 Restores a system from a file.

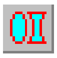
 Saves system data to existing file.

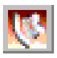
 Saves system data to a new file.

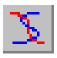
 Option for editing configuration data such as aperture, field, wavelength.

 Invokes the surface spreadsheet editor.

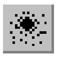
 Lists prescription data of the optical system.

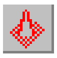
 Displays the lens layout plot (Y/Z-plane).

 Interface to POV-Ray for a rendered 3-D view of the optical system.

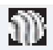
 Displays ray aberrations (fan) across pupil in tangential and sagittal direction.

Displays optical path (OPD) aberrations (fan) across pupil.

 Displays the spot diagram.

 Calculates diffraction point spread function (PSF).

 Calculates a grey-shaded picture of the PSF.

 Calculates a pseudo interferogram from the wavefront.

Calculates diffraction MTF vs. field.

Calculates diffraction MTF vs. spatial frequency.

Calculates diffraction MTF vs. defocus.

Edit optimization variables and constraints.

OPT PAR	Edit optimization settings and parameter.
OPT	Run the (local) optimization.
GO	Run the global optimization.
GOV	Global optimization viewer
AF	Autofocus. Find best focus at selected fields and wavelengths.
ZOO	Edit zoom/multiconfiguration data.
TOL	Edit tolerances.
BPM	Edit (physical optics) beam propagation parameter .
RAY	Generate rays for lens layout plot.
MHT	Calculate the maximum required surface heights according to the system definition (includes fixed apertures, vignetting, etc.)
MAC	Macro editor
	Edit thin film construction parameters.
REF	Calculate thin film reflectance properties.
TRA	Calculate thin film transmittance properties.
?	Invokes the HTML help system.

3.1.4 Command Line

In addition to using menus, *OpTaliX* can be run using commands only. Two command lines are provided, one in the footer of the text output window and one in a floating window that can be freely located on the desktop (i.e. it is not confined in the *OpTaliX* main window) as shown in Fig. 3.3. They are fully equivalent in functionality. The commands and their detailed syntax are described in the reference manual and in the online help system. A convenient way to learn commands is to use the history window in conjunction with the menus. Most actions from menus are reported in the history window.

It is normally not necessary to put the cursor into either command line to type commands. The program detects keyboard input of valid characters and changes the focus automatically to the command line in the text window.

Entering and changing data is accomplished by a free format command syntax which is similar to CODE-V commands in many (but not all) respects. All commands are case insensitive and parameters must be separated at least by one blank. Multiple blanks are treated as a single blank. Most of the commands accept the "question mark" symbol `?`, which invokes a dialog based entry of additional parameter.

Commands can include numeric expressions for parameters such as `THI s3 sqrt(2)+1`. Several commands can be given on a single line, separated by semicolons. Each command line can be up to 256 characters.

Many commands accept parameters for surfaces, field, wavelength, zoom positions, rays, coefficients, pupils, etc. Accepted parameters are shown below.

`si..j`	Surface range (surfaces i to j)
`fi..j`	Field range (field numbers i to j)
`wi..j`	Wavelength range (color numbers i to j)
`ri..j`	Ray range (ray number i to j)
`zi..j`	Zoom range (zoom position i to j)
`ci..j`	Coefficients (range i to j)
`pi..j`	Pupil (surface aperture) range i to j

For example, `s3..4` denotes surfaces 3 to 4. If a command is entered with no arguments, default values will be used where possible. Some commands use settings in the configuration dialog. For example, the spot diagram uses the values for field and wavelength as defined in the configuration.

The following examples give a first overview about the syntax of command entries. A more detailed description with all options is given in the reference manual. Text right to the exclamation mark ''!'' are comments and do not belong to the command.

```
rdy s1..3 10.0   ! sets the Y-radii of surfaces 1 to 3 to 10.0
yan f4 2.5       ! sets thy Y-angle of fields 2 to 4 to 2.5 (degree)
psf f1           ! calculate diffraction point spread function at field 1
```

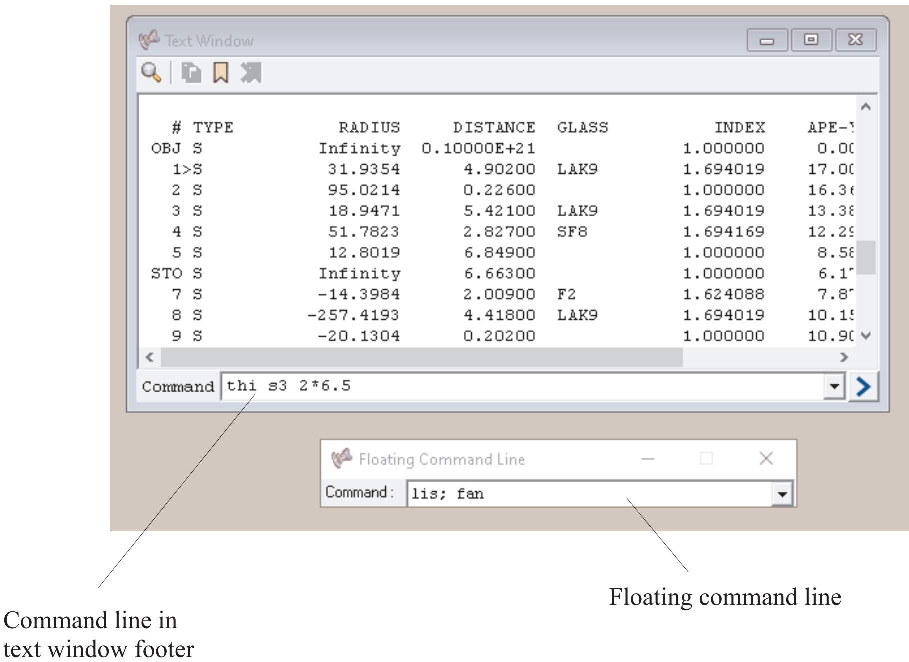


Figure 3.3: Command line at the bottom of the text output window and showing the usage of numeric expression for parameters. Note the use of several commands on a single command line, separated by semicolons as shown in the floating command window.

3.1.5 Status Bar

The status bar is found at the bottom of the main window and contains information about (from left to right):

Grid The ray grid used for performance evaluation. The rays are defined in an N by N grid over the entrance pupil. The higher N is, the more accurate will be the calculations, however, the longer will be the execution time. the ray grid is defined in the configuration (command : `EDI CNF`, from the menu: Edit –> Configuration).

OUT Designates the output unit for text and numerical performance results.

GRA Designates the output unit for graphics.

POS Shows the currently selected zoom/multiconfiguration position.

Path Displays path and filename of the currently loaded optical system.

3.2 Graphics Windows

For each type of graphics a separate window is opened, which may be individually changed in size and may be arbitrarily placed on the screen.

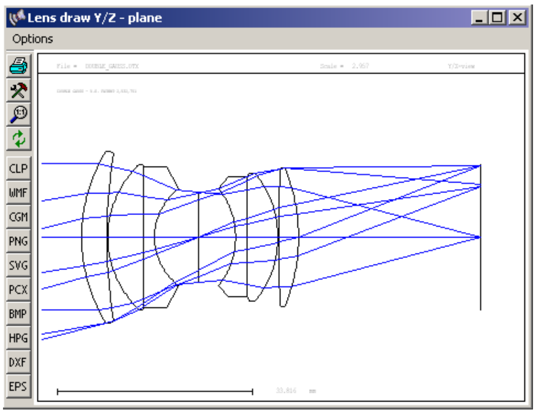


Figure 3.4: Graphics window

3.2.1 Graphics Window Toolbar

A toolbar is associated to each graphic window, which allows optional actions, such as printing, changing plot parameters, zoom, unzoom, export graphics or update the graphics window. The meaning of each icon is as follows:

Modify plot parameter, such as plot scale, aberration scale, etc. This tool opens individual dialog boxes for each graphics window.

Print graphics to default printer. This may also be a network printer.

Unzoom the plot, i.e. show the full plot area.

Update graphics, for example if design data were changed. See also the double-click option described in section 3.2.3.

The contents of graphics windows can also be exported to other established formats. This is accomplished by the following icons:

CLP Copy graphics to clipboard, see also section 3.3 below.

WMF Export graphics to **W**indows **M**etafile **F**ormat.

CGM Export graphics to **C**omputer **G**raphics **M**etafile format.

PNG Export graphics to **P**ortable **N**etwork **G**raphics format.

SVG Export graphics to **S**calable **V**ector **G**raphics format

PCX Export graphics to Paintbrush format

BMP Export graphics to Windows bitmap format

HPG Export graphics to **H**ewlett **P**ackard **G**raphics **L**anguage format

DXF Export graphics to AutoCAD **D**rawing **Ex**change **F**ormat

EPS Export graphics to **E**ncapsulated **P**ost**S**cript format.

3.2.2 Graphics Window Zoom

Portions of the whole plot area may be enlarged by dragging a rectangle with the mouse around the area of interest. In a graphics window, click on one corner, hold the left mouse button down and drag a rectangle over that area you want to have enlarged (Fig. 3.5). releasing the mouse button will instantly display the enlarged plot.

3.2.3 Graphics Update

Graphics windows are automatically updated when required. There is normally no user interaction required. However, plots with high underlying computational effort (such as MTF, PSF, etc.) must be updated on user request, because otherwise the program response may slow down unnecessarily.

Graphics update is accomplished by either clicking on the update icon in the toolbar to the left of each graphics window, or by simply double-clicking anywhere in the frame of a graphics window.

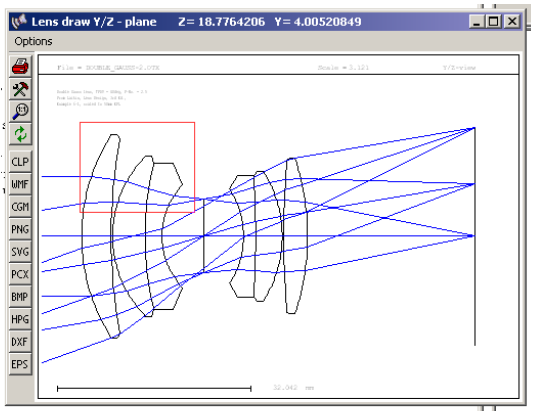


Figure 3.5: Enlarging a plot area by dragging a rectangle with the mouse.

3.3 Clipboard Support

Windows clipboard is supported for text and graphics. Text in either the text output window or the history window can be copied by marking the desired test portion black. To do this, click on the upper left corner of your selection, hold the left mouse button and drag the mouse to the lower right corner of the selection. Then release the left mouse button and press **Ctrl-C** on the keyboard. The text is then available in the clipboard for other programs.

Graphics can also be copied to the clipboard from any open graphics window. Select *Options –> Copy to Clipboard* from the menu attached to each window or simply click on the 

icon to the left of each graphics window . The vector graphics is placed in the clipboard in standard Windows metafile format.

3.4 The Surface Editor

The surface editor is a tabbed dialog which contains several spreadsheets for editing surface parameter. The surface editor is invoked from the main menu *Edit –> Surface Data* or by clicking on the  icon in the toolbar or by entering `EDI SUR` in the command line. The surface parameter are grouped in several tabs as shown in Fig. 3.6):

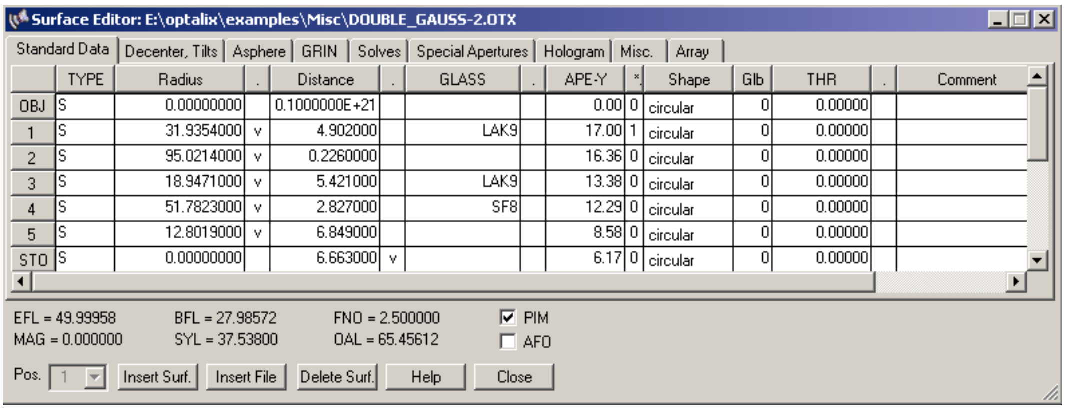


Figure 3.6: Surface spreadsheet editor.

Standard Data This spreadsheet contains the most used data such as radius of curvature, thickness, glass, circular apertures, pickup's, surface comments, surface type, etc.

Decenter, Tilts All parameters needed to decenter and/or tilt surfaces are found in this spreadsheet. Special tilt modes may also be defined here.

Asphere Contains deformation coefficients for rotationally symmetric aspheres and for toroidal surfaces.

GRIN Allows definition of the gradient index (GRIN) model and also contains decenter/tilt data of the index profile.

Solves Solves allow direct control of paraxial parameters such as incidence angle, paraxial ray angle or ray height, aplanatic condition, etc.

Special Apertures Special apertures are all apertures, which are not circular. Defines rectangular, elliptical, circular and polygon aperture shapes, which can be logically combined.

Hologram Coefficients for defining holograms and gratings.

Misc Miscellaneous surface data.

Array Data associated to array elements, such as spacing and size of array cells (channels), etc.

Also note the dialog fields in the bottom of the spreadsheet for selecting zoom position, paraxial image position (PIM) and for inserting and deleting surfaces.

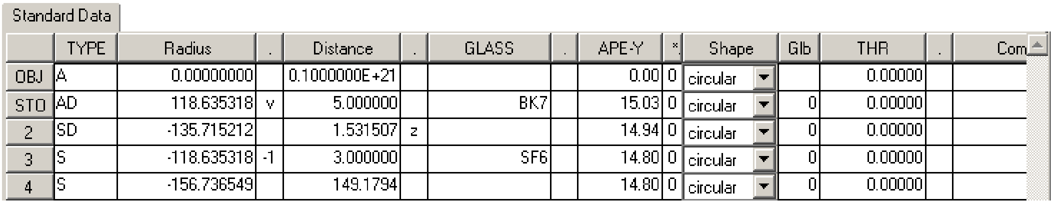
Navigating around in a spreadsheet can be accomplished with the mouse or keyboard. Mouse navigation is quite simple:

- Click in a cell to select it,
- Click in a row or column heading to select a whole row or column,
- Use the scroll bars to move around in the spreadsheet,

- **A value is only accepted if the Enter or Return key is pressed, or if the cursor is placed in another cell !**

The following sections give a concise description of each column in the tabbed spreadsheets.

3.4.1 Standard Data


Standard Data

	TYPE	Radius	.	Distance	.	GLASS	.	APE-Y	*	Shape	Glb	THR	.	Com
OBJ	A	0.00000000		0.1000000E+21				0.00	0	circular		0.00000		
STO	AD	118.635318	v	5.000000		BK7		15.03	0	circular	0	0.00000		
2	SD	-135.715212		1.531507	z			14.94	0	circular	0	0.00000		
3	S	-118.635318	-1	3.000000		SF6		14.80	0	circular	0	0.00000		
4	S	-156.736549		149.1794				14.80	0	circular	0	0.00000		

The tab ''Standard Data'' contains often used data such as radii of curvature, thicknesses, glasses, apertures, etc. Extra columns (labelled by a dot) are located to the right of each of the curvature, thickness, material and global reference thickness columns. They accept a surface number to pickup a surface to a preceding surface, to make a surface radius/thickness/material variable in optimization or to indicate a solve on a particular item. The accepted entries to the "dot-columns" are:

v	makes the designated item variable in optimization (single position variable),
z	makes the designated item a zoom variable in optimization,
integer number	indicates a pickup from a preceding surface. A surface number with negative sign picks the surface with opposite curvature or distance.
s	indicates a solve (display only). Solves are set in the solve tab.

The meaning of the column headings is as follows:

Type Designates the type of surface. This is a string of up to four surface descriptors (a character), which can be almost arbitrarily combined. Surface types are categorized into obligatory and optional ones, according to the following table:

Obligatory Surface Types		Optional Surface Types	
S	Spherical surface	D	Decentered and/or tilted surfaces
A	Aspheric surface.	M	Mirror
L	Lens module (ideal lens)	G	Grating surface
X	"No-raytrace" surface. Only transforms surface coordinates without actually tracing rays to this surface.	H	Holographic surface
U	User defined surface	F	Fresnel Surface
		I	Gradient Index
		N	Non-sequential surface, must be used in combination with surface type ''D''
		P	Light Pipe, step index fiber. The cone angle of tapered pipes/fibers is defined by the semi-apertures of the end surfaces
		R	Array of identical surface elements
		T	Total internal reflection (TIR) surface
		Z	Zernike surface
		C	rotationally symmetric spline surface deformation
		W	2-dimensional surface deformation, given as gridded data
		E	pure 2-dimensional spline (non-symmetric), no base surface. In preparation.

For example, SDM defines a surface of spherical shape, it is decentered (and/or tilted) and is reflecting (mirror).

Radius Radius of curvature. It is positive if the center of curvature lies to the right of the surface, otherwise it is negative. This rule is independent of the direction of the light.

Distance The separation of two consecutive surfaces along the optical axis. It is positive if the next surface lies to the right of the current surface, otherwise it is negative. Distances following a mirror are also negative.

Glass Name of a predefined optical material (glasses, plastics, liquids, etc).

APE-Y Semi-aperture diameter.

***** This column indicates if a surface aperture is checked. Only values of 0 (unchecked) or 1 (checked) are accepted. If checked, rays outside the surface aperture are blocked. Note that the stop surface is always a checked surface, irrespective of the 0/1 setting in the * column.

Shape The aperture shape can be either circular (the default), elliptical, rectangular or a polygon.

Glb Global referencing. Refers a surface to the local surface normal and vertex of a *preceding* surface.

Comment Surface note.

Coating Attach a multi-layer coating to a surface by specifying a coating file name (max. 8 characters).

3.4.2 Decenter, Tilts

All parameters for defining decentered and/or tilted surfaces can be entered from this tab. Extra columns (labelled by a dot) are located to the right of the XDE, YDE, ZDE, ADE, BDE, CDE columns. The accepted entries to the ”dot-columns” are:

- **v** parameter is variable in optimization (for single position variable),
- **z** parameter is variable in a zoom (multi-configuration).
- **number** an integer number defines a pickup of the designated parameter from another surface.

Decenter, Tilts

	THR	TLM	SEQ.	Pik	XDE	..	YDE	..	ZDE	..	ADE	..	BDE
OBJ	0.00000	DAR	XYZABC	0	0.000000		0.000000		0.000000		0.000000		0.0
1	0.00000	DAR	XYZABC	0	0.000000		0.000000		0.000000		0.000000		0.0
2	0.00000	NAX	XYZABC	0	2.500000	v	0.000000	v	0.000000		15.00000	v	0.0
3	0.00000	DAR	XYZABC	0	0.000000		0.000000		0.000000		0.000000		0.0

THR Reference thickness. This is the axial separation (thickness) of a surface which is referenced to a preceding surface. Unlike ’’normal’’ thicknesses, a reference thickness is always defined before the surface.

TLM Tilt mode. It defines the coordinate system for subsequent surfaces. Three tilt modes are available:

- `DAR` Decenter and return. The coordinate system is returned to the optical axis after a surface tilt/decenter. This way, the optical axis is not altered.
- `NAX` New axis. As the name implies, the local surface normal defines the new axis for all subsequent surfaces.
- `BEN` Bend surface. The optical axis follows the law of reflection. Can be used only in conjunction with a mirror.

SEQ Tilt sequence. It describes the order of tilts/decenters. The letters X, Y and Z denote decenters. A, B, and C are the tilts about the X-axis, Y-axis and Z-axis. Thus, the string XYZABC would apply the coordinate transformation of a surface in the order X-decenter, Y-decenter, Z-decenter, rotation about X-axis, rotation about Y-axis, rotation about Z-axis. This is also the default.

Pik Indicates a tilt pickup from another arbitrary surface. A tilt pickup in the ’Pik’ column is a *group pickup*, i.e. applies to all tilt parameters on the designated surface(s) A tilt pickup in one of the ”dot columns”, i.e. right to a XDE, YDE, ... column, indicates an *individual* pickup. This will disable any group pickup on that surface, if existent.

A tilt pickup with negative sign picks up the corresponding tilt/decenter parameter with negative sign.

XDE X-decenter

YDE Y-decenter

ZDE Z-decenter

ADE Tilt about X-axis in degree (α - tilt)

BDE Tilt about Y-axis in degree (β - tilt)

CDE Tilt about Z-axis in degree (γ - tilt)

3.4.3 Asphere

Parameters for aspheric surfaces are entered by clicking on the third tab. Aspheres are described by a polynomial function, which is rotated about the local Z-axis. The definition includes conic surfaces (parabola, ellipse, hyperbola) and toroids. Extra columns (labelled by a dot) are located to the right of the A, B, C, D, E, F, G, H columns. The accepted entries to the "dot-columns" are:

v	the designated parameter is variable in optimization (for single position systems),
z	the designated parameter is a zoom (multi-configuration) variable in optimization.
number	an integer number defines a pickup of the designated parameter from another surface.

Asphere

	Asph.Type	Pik	K (Conic Const.)	.	A (h^4)	.	B (h^6)	.	C (h^8)	.	D (h^10)	.	E (h^12)	.
OBJ S	even, 18th		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000	
1 S	even, 18th ord		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000	
2 S	even, 18th ord		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000	
3 S	even, 18th ord		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000	
4 S	even, 18th ord		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000	

Asph.Type Selects the definition of higher order coefficients. The selectable options are ''ODD'', ''ODD30'', ''EVEN'', ''XYP'', "anamorphic", "Cylinder", "Qcon", "Qbsf", "User defined" and "Xfreeform" coefficients sets.

K The conic constant K describes surfaces of conic sections.

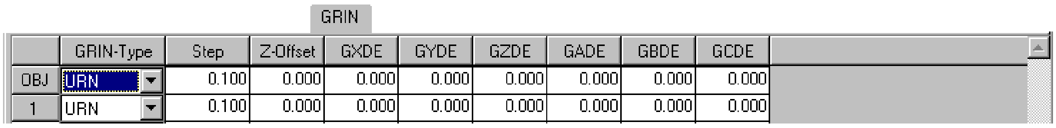
		K	$<$	-1	Hyperbola
		K	$=$	-1	Parabola
-1	$<$	K	$<$	0	Ellipse at major axis (prolate ellipse)
		K	$>$	0	Ellipse at minor axis (oblate ellipse)
		K	$=$	0	Sphere

A, B, C, D, E, F, G, H Higher order polynomial coefficients. The meaning of A,B,C,D,E, ... depends on the asphere type (`ATY EVEN|ODD9|ODD30|XYP|QCN|QBF|XFF`).

RDX Radius in X/Z-plane. If non-zero, the surface is toroidal (different curvatures in X- and Y-direction).

3.4.4 GRIN

Gradient index (GRIN) properties. Although they basically describe material properties, the data are attached to a surface.



GRIN-Type The gradient index type describes the underlying equation of the index profile. The following profiles are available:

URN	University of Rochester gradient. A mixed gradient in axial and radial direction
SEL	Selfoc^{TM} gradient from Nippon Sheet Glass company (NSG)
LPT	Gradium^{TM} axial profile from LightPath
AXG	Axial gradient
LUN	Luneberg gradient
SPG	Spherical gradient
MAX	Maxwells Fisheye
GLC	radial gradient from Gradient Lens Corporation (GLC)

Step The integration step length along the optical path used during raytrace. The smaller the step, the more accurate will be the result, however, the more computing intensive it will be.

Z-Offset Defines the location of the surface vertex wrt. the profile. Required only for axial gradients (LPT-Gradium).

GXDE X-decenter of the gradient index profile wrt. the local surface vertex.

GYDE Y-decenter of the gradient index profile wrt. the local surface vertex.

GZDE Z-decenter of the gradient index profile wrt. the local surface vertex.

GADE Tilt of the gradient index profile about the local surface X-axis.

GBDE Tilt of the gradient index profile about the local surface Y-axis.

GCDE Tilt of the gradient index profile about the local surface Z-axis.

Coeff. Opens a dialog to edit user-defined GRIN coefficients. This option can only be selected if the designated surface is a GRIN surface, the glass name is ''GRIN'' and the gradient index profile type (`GIT`) is either `URN`, `LPT` or `UDG`.

MXG Maximum number of iteration steps in GRIN medium. Terminates ray iteration after a maximum number of iterations has been reached. Enter ''MXG 0'' (without the quotes) to disable limit checking. Note that `MXG 0` will not set the number of iteration steps to infinity but will terminate ray iteration internally after 5000 steps to avoid infinite loops (for example when GRIN coefficients are improperly set). Any number except 0 defines a hard limit to stop ray iteration. Thus, if you need more than 5000 steps in a medium it must be explicitly set, for example by `MXG 20000`.

3.4.5 Solves

Defines solve parameters. Solves allow direct control of paraxial properties. Conditions for specifying a solve are, for example, holding the paraxial ray angle, the paraxial ray height or a certain paraxial ray incidence angle to a specified value. Solves will keep these requirements satisfied.

Solves

	Solve-Type	Param. 1	Param. 2	Solve-Type	ET	Semi-diam.
OBJ	UMY -ray angle Y	0.0000	0.0000	ET -edge thicknes	0.0000	0.0000
1	none	0.0000	0.0000	none	0.0000	0.0000

Solve-type The solve-type defines which *paraxial* requirement will be fulfilled. Available types are ray angle, ray height, incidence angle, aplanatic, edge thickness. Two solve types can be applied to a surface.

Param.1 The first parameter. It depends on the solve type. The column heading will be updated corresponding to the selected solve type to aid the user in entering the correct value.

Param.2 The second parameter is only required for an edge-thickness solve. It specifies the radial height at which a certain edge-thickness shall be maintained.

3.4.6 Special Apertures

Special apertures are all apertures, which are not circular. Defines up to 10 basic aperture shapes (rectangular, elliptical, circular and polygon) on a surface. The basic aperture shapes may be logically combined by OR and AND operators.

Each aperture element may be transmitting or obstructing. Each aperture can be decentered in X- and Y-direction from the local surface vertex and it can be rotated. Basic apertures may be logically combined by `.and.` respectively `.or.` operators.

In the special apertures dialog (see Fig. 3.7) select the surface from the drop-down list to the left of the dialog. The current aperture data for this surface will then be updated in the table. At least one basic aperture is required, which will be circular in most cases.

Basic apertures must be activated to be included in ray trace. Check the appropriate box in the ''On'' column. If there is more than one basic aperture on a surface, the aperture data must be entered consecutively into the table.

To edit polygon shapes, first select ''polygon'' in the ''Shape'' column. The corresponding button in the ''Polygon'' column (rightmost in the table) will be activated. Click on this button and enter the polygon vertices for this basic aperture.

A more detailed example on defining special apertures is given in sect. 4.4 (page 41).

3.4.7 Hologram

Commonly, holographic surfaces are also denoted as *diffractive* surfaces. The optical properties of a holographic surface are based on diffraction at the effective grating spacing seen at the local intersection point of a ray. A grating is handled in *OpTaliX* as a special case of a holographic surface.

Holographic surfaces are highly dispersive. Several types of diffractive (hologram) surfaces are available in *OpTaliX* .

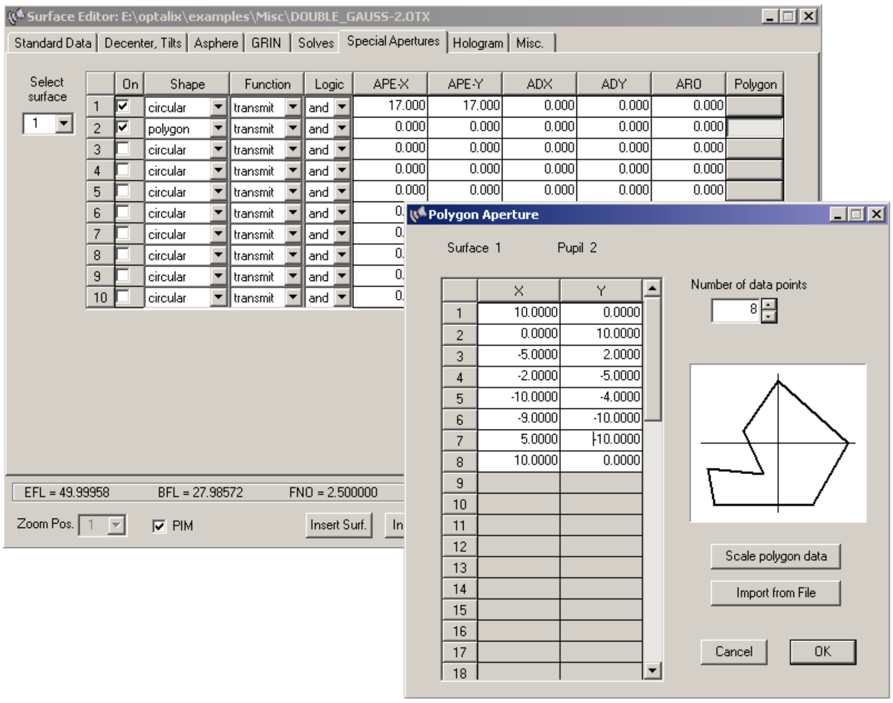


Figure 3.7: Editing special apertures. A separate dialog is invoked for defining vertices of polygon apertures.

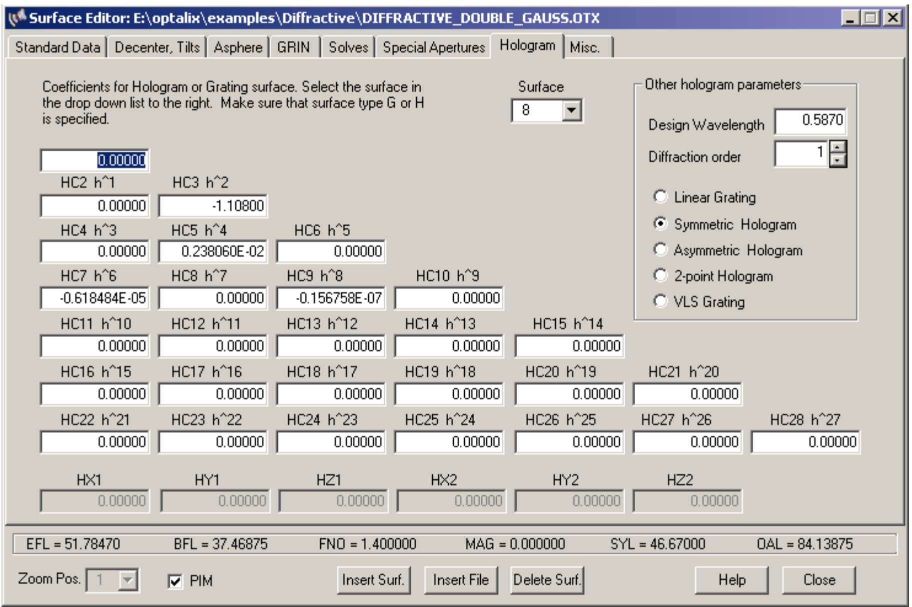


Figure 3.8: Editing hologram coefficients.

- Linear grating,
- Variable linear spacing (VLS) grating,
- Optical hologram, formed by interfering two beams of light,

- Computer-generated holograms (CGH) with a user specified radial symmetric phase distribution,
- Computer-generated holograms (CGH) with a user specified asymmetric (two-dimensional) phase distribution,
- ''Sweatt'' model using a very high index of refraction.

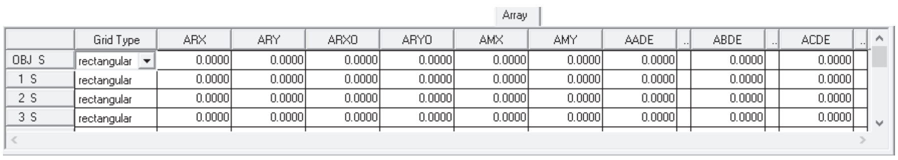
The hologram type can be selected in the right part of the dialog. Each hologram surface requires specification of the design wavelength and diffraction order.

Note that the hologram coefficients describe the phase change of the wavefront. The first derivative is the local grating constant at a particular point on the hologram surface. For linear gratings, the first derivative is a constant. Thus, in that special case, the grating constant (i.e. grooves/mm) is identical to the hologram coefficients. This makes it particularly easy to define gratings.

3.4.8 Array

The array surface arranges optical elements (surfaces) in a regular grid, i.e. they are repeated many times at specified X/Y locations with respect to the local coordinate of a surface, denoted hereafter as *array cells* or *channel surface*.

The individual lens or surface assemblies may be regarded as *cells* or *channels*. The channel surface encompasses all of the channels in the array. The aperture limits of the array surface are defined by the `AMX`, `AMY` parameters. Depending on the aperture dimensions and the cell/channel spacings (`ARX`, `ARY`), some channels (array elements) may be truncated. Individual channels are distributed in a uniform rectangular grid over the channel surface, as indicated in Fig. 3.9 and 3.10. A hexagonal arrangement of the array cells is accomplished by selecting the grid type (first column) to ”hexagonal” as shown in Fig. 3.11. Then all cells in odd numbered rows are shifted such that the center points of the cells form triangles instead of rectangles. The channel centers are located at (local) X/Y coordinates defined by the X-spacing (`ARX`) and Y-spacing (`ARY`).


Array

	Grid Type	ARX	ARY	ARXO	ARYO	AMX	AMY	AADE	ABDE	ACDE
OBJ S	rectangular	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
1 S	rectangular	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2 S	rectangular	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
3 S	rectangular	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 3.9: Definition of array cells (channels).

ARX X-spacing of array channels/cells.

ARY Y-spacing of array channels/cells.

ARXO X-offset of entity of array channels with respect to local surface coordinate system.

ARYO Y-offset of entity of array channels with respect to local surface coordinate system.

AMX $\pm$ limit for grid in X-direction

AMY ± limit for grid in Y-direction

AADE α-tilt angle (in degree) of each array cell.

ABDE β-tilt angle (in degree) of each array cell.

ACDE γ-tilt angle (in degree) of each array cell.

The definition of array cells is also illustrated in Fig. 3.10 and Fig. 3.11.

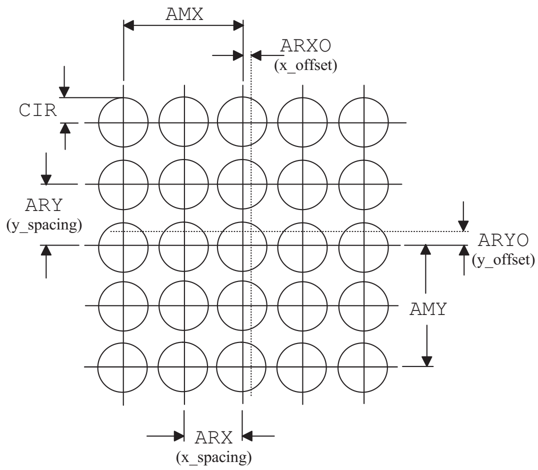


Figure 3.10: Definition of array cells (channels) for a square regular grid. The dashed lines indicate the vertex of the base surface.

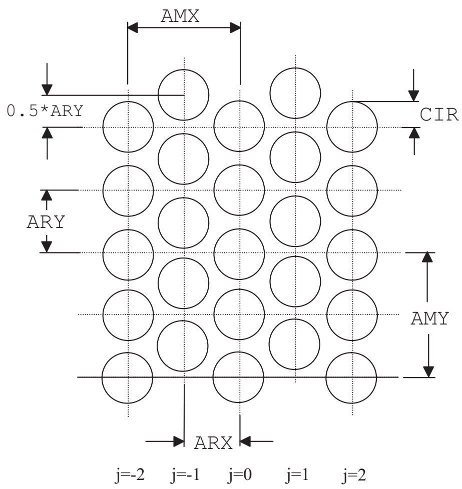


Figure 3.11: Hexagonal arrangement of array cells. All odd numbered columns are shifted (staggered) in Y-direction by 0.5*ARY.

3.5 Entering a new System

Our first and relatively simple example will be an achromatic doublet, which we will set up from scratch. This requires input of surface data as well as configuration data. Note, that in the terminology used throughout the manual, *configuration data* are data that pertain to the whole lens or describe its conditions of use. For example, typical configuration data, among others, are aperture, field of view and wavelength. The doublet will have a focal length of 200mm, a diameter of 30mm and a field of view of ±1 degree. The object is assumed at infinity.

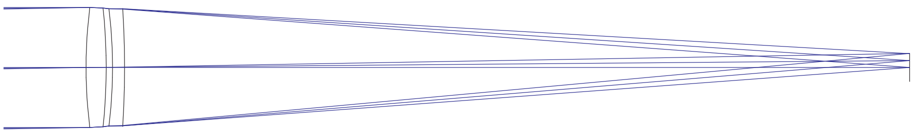
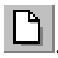
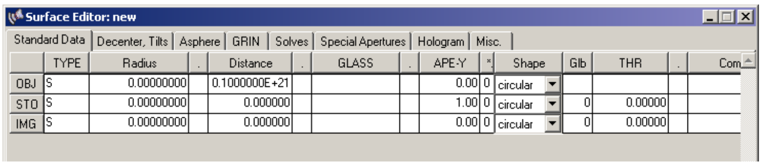


Figure 3.12: Example achromatic doublet.

The course examples will be given in two forms, first by using menus and dialogs and then using commands only.

3.5.1 Menu Entry

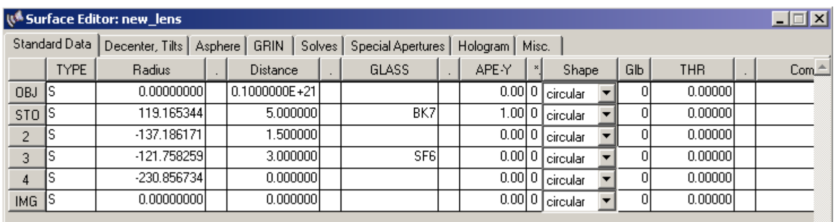
We will create a new lens, thereby erasing all design data of the previous system. This is accomplished from the file menu *File –> New* or by clicking on the toolbar icon .
Next, surface data are entered. This is accomplished by invoking the surface spreadsheet editor (if not already open). Select *Edit –> Surface Data* from the main menu or click on the toolbar icon . The table contains three rows, corresponding to the object surface labelled `OBJ`, the stop surface labelled `STO` and the image surface labelled `IMG`.



	TYPE	Radius	.	Distance	.	GLASS	.	APE-Y	*	Shape	Glb	THR	.	Com
OBJ	S	0.00000000		0.1000000E+21				0.00	0	circular				
STO	S	0.00000000		0.000000				1.00	0	circular	0	0.00000		
IMG	S	0.00000000		0.000000				0.00	0	circular	0	0.00000		

For a doublet, the system must have 6 surfaces in total, four for the refracting surfaces and two for object and image surface respectively. In our example we chose the first optical surface to be the stop surface and three surfaces must be added to the system. Thus, we place the cursor into the third row (labelled `IMG`) and insert three surfaces by clicking on the ''Insert Surf.'' button underneath the table. We do this three times.

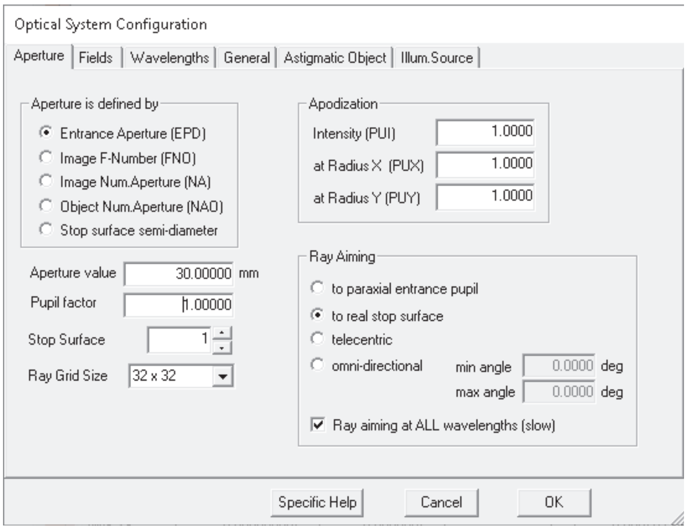
As a minimum, radius of curvature, axial separations and glasses must be entered. For the moment we leave the surface apertures in the APE-Y column zero. They will be automatically calculated later. The surface spreadsheet now should look like this:



	TYPE	Radius	.	Distance	.	GLASS	.	APE-Y	*	Shape	Glb	THR	.	Com
OBJ	S	0.00000000		0.1000000E+21				0.00	0	circular	0	0.00000		
STO	S	119.165344		5.000000		BK7		1.00	0	circular	0	0.00000		
2	S	-137.186171		1.500000				0.00	0	circular	0	0.00000		
3	S	-121.758259		3.000000		SF6		0.00	0	circular	0	0.00000		
4	S	-230.856734		0.000000				0.00	0	circular	0	0.00000		
IMG	S	0.00000000		0.000000				0.00	0	circular	0	0.00000		

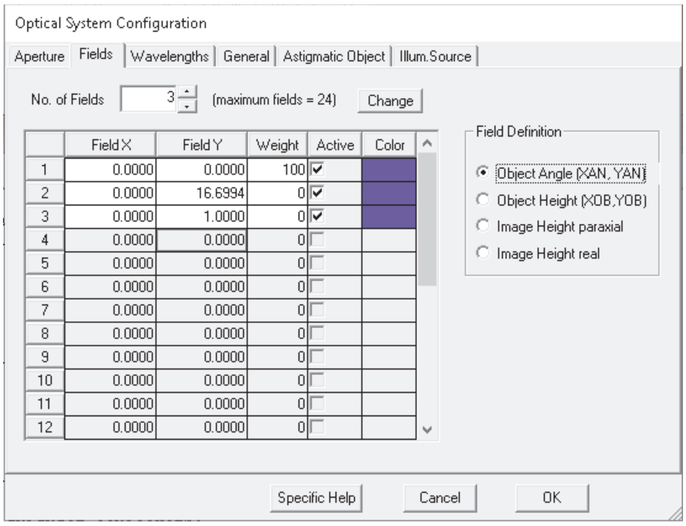
We also wish to adjust the position of the focal plane, which is defined by the last distance (on surface 4). This distance is always the separation from the last surface to the image surface. It can be automatically adjusted, depending on the status of PIM (paraxial image). See also the check box ''PIM'' in the lower part of the surface spreadsheet editor as shown in Fig. 3.6. If PIM is checked, the image surface will always be moved to the location where the *paraxial* image is. Thus, the last distance will change on modification of surface parameters (radius, distance, glasses, etc). If PIM is unchecked, the separation of the image surface to the last surface will not be automatically updated. Hence, the last distance in the spreadsheet editor will not change and the user is responsible to place the image surface to a suitable position.

In the next step we will define the operating conditions in the configuration dialog. It is invoked from the main menu *Edit –> Configuration data* or by clicking on the toolbar icon . The dialog contains several tabs to which the operating data are grouped.

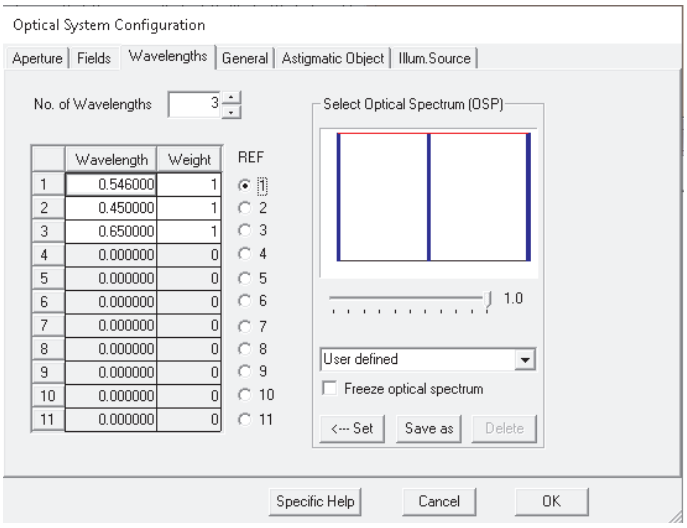


The ''Aperture'' tab offers several methods of defining the aperture. We click on the radio button ''Entrance aperture (EPD)'' and enter the desired value (30.0 mm) in the numeric field below. We leave the stop at surface 1. Since the doublet is used only near to the optical axis the stop position is uncritical.

The field of view is specified in the ''Field'' tab. We divide the maximum field (± 1 degree) into three field angles at 0, 0.5 and 1 degrees. Note, that the entered values are always semi-field angles measured from the optical axis in X- and Y-direction separately. We enter the number of fields (3) and the field angles in the ''Field Y'' column. Since the object is at infinity, the only meaningful definition of the field data is by object angle. This is the default and no further changes are necessary.



One wavelength (546nm) is defined in the ''Wavelengths'' tab. Please keep in mind that all wavelengths in *OpTaliX* are specified in microns, so we enter 0.546 for a wavelength of 546nm, the green mercury (Hg) line. We will add two other wavelengths, 450nm and 650nm, to have three wavelengths which approximate roughly the visible spectral range. The number of wavelengths must be increased to three in the list box above. The reference wavelength is the wavelength at which the paraxial properties are calculated. We will not change this setting and leave it at the first wavelength (546nm). The weights are integer numbers between 0 and 100 and are used to simulate a relative spectral distribution. The absolute value of the weights is not important. Entering 1 for all wavelengths results in a uniform (flat) spectral distribution.



All necessary operating parameters have been defined and we and close the configuration dialog. The entry of the optical system is almost finished and we can print the lens description to check if the paraxial data are as expected. Select *List –> Surfaces* from the main menu or click on the toolbar icon . The output should be as follows:

```
FILE = new_lens
```

```
Remarks:

Wavelength :      0.54600     0.45000     0.65000
Weight     :            2           1           1
REF =  1

XAN        0.00000       0.00000       0.00000
YAN        0.00000       0.50000       1.00000
FWGT           100           100           100

PIM = no
SYM = yes
EPD =  30.0000

  # TYPE        RADIUS     DISTANCE  GLASS         INDEX    APE-Y AP CP DP TP MP GLB
OBJ>S         Infinity  0.10000E+21             1.000000     0.00  C  0  0  0  0   0
STO S         119.1653      5.00000  BK7         1.518726    15.00  C  0  0  0  0   0
  2 S        -137.1862      1.50000             1.000000     0.00  C  0  0  0  0   0
  3 S        -121.7583      3.00000  SF6         1.812665     0.00  C  0  0  0  0   0
  4 S        -230.8567      0.00000             1.000000     0.00  C  0  0  0  0   0
IMG S         Infinity                          1.000000     0.00  C  0  0  0  0   0

Paraxial Data:
   EFL                       200.00000      SEP (Entr.Pup.Loc.)       0.00000
   BFL                       194.18610      EPD (Entr.Pup.Dia.)      30.00000
   FNO (F-Number)              6.66667      SAP (Exit Pup.Loc.)      -6.48230
   MAG (Magnification)         0.00000      APD (Exit Pup.Dia.)      30.10026
   DEF (Defocus)               0.00000      SH1 (Princ.Plane 1)       0.66618
   NAO (Num.ape.object)        0.00000      SH2 (Princ.Plane 2)      -5.81390
   NA  (Num.ape.image)        -0.07479      SYL (System Length)       9.50000
   S0                   -0.1000000E+21      OAL (S1->Image)           9.50000
   OID (Object->Image)     0.10000E+21      PRD pupil relay dist      3.01770
```

The focal length (`EFL`) is 200 mm as expected. The remaining parameter to be defined are the surface apertures for surfaces 2 - 5. Instead of typing the values manually into the spreadsheet, we let the program calculate it. The option `MHT` determines the maximum required surface heights on the basis of the previously entered fields and system aperture. This option is found in the main menu under *Tools –> Set maximum heights* or by clicking on the toolbar icon MHT.

As a final step we plot the lens layout, which is invoked from the main menu *Display –> Lens Draw Y*. It should be identical to Fig. 3.12.

3.5.2 Command Line Entry

This section repeats the steps of the previous section, but now using only commands. Text after the exclamation mark ''!'' are comments and are not part of the commands.

```
len                   ! Creates a new system. All previous design data are deleted.
ins s1..3             ! Inserts three surfaces before the last surface
```

We will now enter surface data such as radius of curvature, thicknesses and glasses:

```
rdy s1 119.1653       ! Sets radius of curvature of surface 1.
thi s1 5              ! Thickness after surface 1
gla s1 bk7            ! Glass at surface 1 is Schott BK7.
```

Repeat data entry for surfaces 2 to 4 :

`rdy s2 -137.1862`	
`thi s2 1.5`	
`gla s2`	! The medium after surface 2 is air. Since this is the default for a new lens, the command may also be omitted.
`rdy s3 -121.7583`	
`thi s3 3`	
`gla s3 sf6`	
`rdy s4 -230.8567`	

We are defining system aperture, fields and wavelengths:

`epd 30`	! Sets entrance aperture diameter
`yan 0 0.5 1`	! Defines three field angles at 0, 0.5 and 1 degree.
`wl 0.546 0.45 0.65`	! Defines three wavelengths.
`ref 1`	! Reference wavelength number is 1.
`set mht`	! Sets maximum required surface heights according to the specified aperture and field points.

The lens layout can now be plotted:

vie	! Plots a cross-sectional view of the optical system in the Y/Z-plane.

4

Worked Examples

4.1 Tilted Surfaces Example

In this example we will learn how to work with tilted and decentered surfaces. We will use a Double-Gauss design, which is found in the examples directory under `examples\misc\double_gauss-2.otx`. An alternative (easier) way to load this file is by selecting *File –> Example Library* from the main menu, and in the file selection box browse to the *Misc* subdirectory, select the file double_gauss-2.otx, as shown in Fig. 4.1 , and press OK.

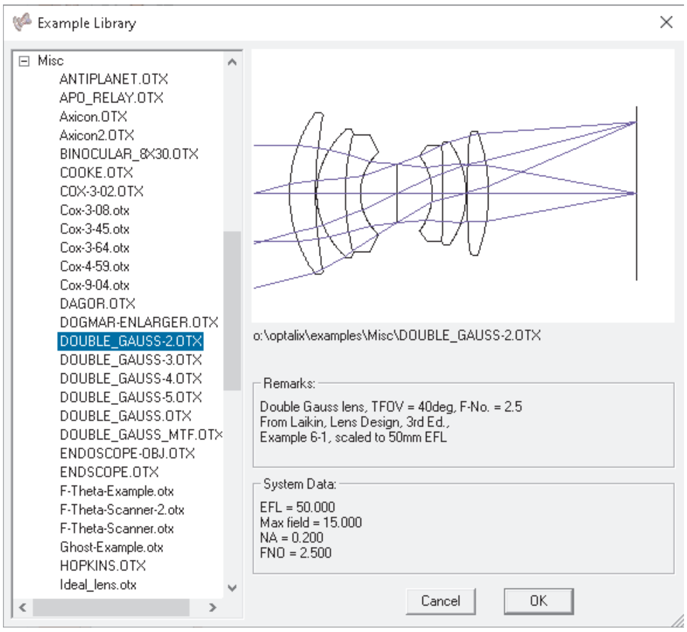


Figure 4.1: Extracting the Double-Gauss-2 example from the examples library.

In this exercise we will insert a 45° fold mirror in the space between last lens surface and image surface. The following figure shows the desired result.

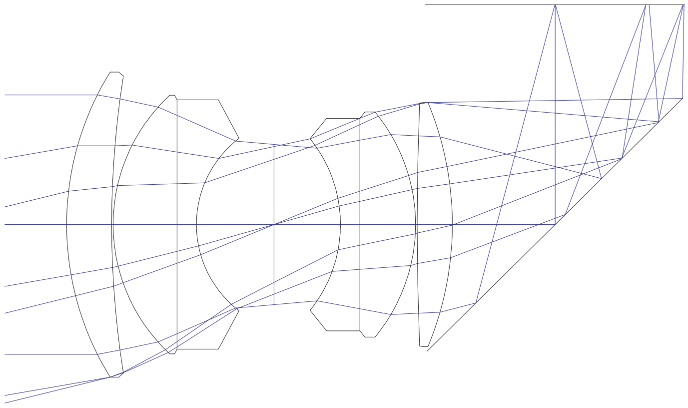


Figure 4.2: Double-Gauss Example with fold mirror.

4.1.1 Menu Entry

The starting design `double_gauss-2.otx` is restored (loaded) as described in the previous section. We may also wish to load the surface spreadsheet editor by selecting *Edit –> Surface Editor* from the menu or directly clicking on the  toolbar icon.
Inserting a fold mirror is accomplished in four steps:

- Inserting an additional surface,
- Tilting the surface,
- Applying a reflecting (mirror) property,
- Setting the appropriate tilt mode (tilting the optical axis).

1. Inserting a surface: The optical system comprises 12 surfaces, including the object and image surface. An additional surface after the last (refracting) surface will be inserted, which will represent the fold mirror. Inserting surfaces are always meant as before the actual surface, that is inserting a surface before the image surface in our example. Thus, in the surface spreadsheet editor, we will place the cursor into any arbitrary column of the image surface, which is the last surface in the spreadsheet, surface 12. This surface is also denoted as ’’IMG’’ in the row labels. Then press the ’’Insert Surf.’’ button, which inserts the surface at the desired position. This is the surface, which we will modify in the following steps.

2. Tilting the surface: The optical system now has 13 surfaces. We change the axial distance at surface 11 in the ’’Standard data’’ tab to 11mm in order to get the fold mirror (surface 11) approximately in the middle of last lens surface and image surface. Some properties of surface 12 must be changed to make it a tilted mirror. First, the surface type of this surface needs to be modified. Looking at the TYPE column in the surface spreadsheet editor, it shows a ’’S’’, meaning a spherical surface. This is definitely correct, since the mirror will be plano (spherical with radius of curvature infinity), however, we must assign reflecting and tilt/decenter properties to this surface. This is accomplished by modifying the surface type to「`SDM`」, where the additional characters ’’`D`’’ and ’’`M`’’ describe the missing properties:

D for decenter/tilt
M for mirror

We enter the tilt angle in the「ADE」column for surface 12.「ADE」is a tilt about the X-axis. Tilt angles are always given in degrees and we will review the sign convention for tilted surfaces:

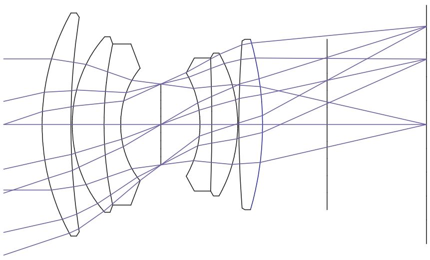
positive sign: tilt is counter clockwise,
negative sign: tilt is clockwise.

A -45° angle will tilt the surface clockwise.

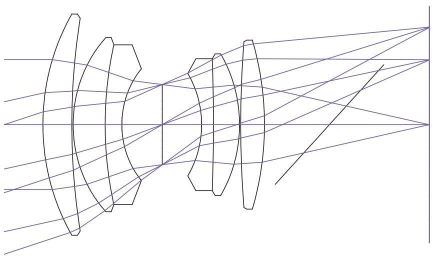
3. Applying a reflective property: The reflecting (mirror) property is solely defined by the surface type (`SUT`). By adding a ”M” character to the surface type a surface becomes reflecting. We have done this already in the previous step by defining `SUT SDM`.

4. Setting the tilt mode: In the surface spreadsheet editor we select the ’’Decenter,Tilts’’ tab and then change in the ’’TLM’’ column at surface 12 the tilt mode (TLM) from ’’DAR’’ to ’’BEN’’. The tilt mode describes how tilted and/or decentered surfaces are handled. The ’’BEN’’ mode deflects the optical axis at a mirror surface according to the law of reflection. Thus it is not required to change the positions of all subsequent surfaces (in our case the image surface), since this is already done by the bend mode (BEN).

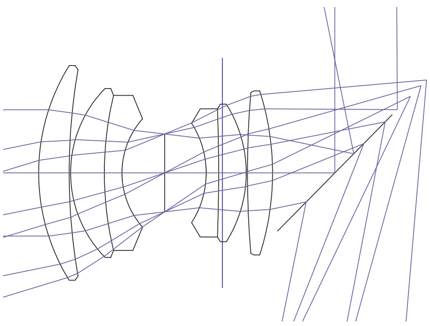
As a final step we apply the `SET MHT` command, which adjusts all surface apertures to the maximum heights (MHT) required by the field definition and the required rays. The resulting lens layout plot (`VIE` command) should now be identical to Fig. 4.2. An illustrated summary of these steps is shown in Fig. **??**.



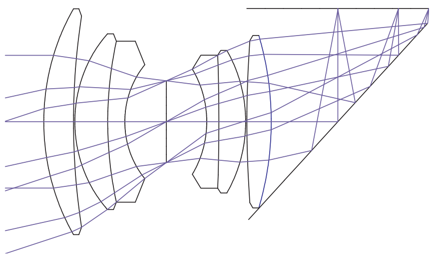
(a) A surface (12) is inserted with a separation of 10mm to surface 11.



(b) The inserted surface is tilted by -45°.



(c) The surface 12 has been made reflecting (mirror, SUT SDM). The optical axis after reflection is not properly aligned.



(d) Setting the tilt modus (TLM) to BEN and adjusting the surface heights using the SET MHT command.

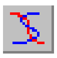
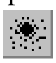
Figure 4.3: Illustration of four essential steps of inserting a tilted fold mirror.

4.1.2 Command Line Entry

It is assumed that the Double-Gauss example, (found in the examples directory under `examples\misc\double_gauss-2.otx`) has already been loaded (restored) into memory. The commands to inserting a fold mirror between last lens surface and image surface are then (see Fig. 4.2 for the result) :

`ins s12`	Inserts a surface before surface 12 (the image surface). The number of the image surface increases to 13.
`thi s11 10`	Changes axial thickness at surface 11 to 10mm.
`sut s12 SDM`	Surface type of surface 12 is spherical (S), decentered (D) and mirror (M).
`ben s12`	Changes tilt mode to `BEN` (optical axis follows law of reflection at a mirror).
`ade s12 -45`	Tilt angle (about X-axis) is 45°. Note the sign convention of tilts: positive values are counter-clockwise, negative values are clockwise.
`set mht`	Sets all surface heights to the maximum heights (MHT) required by the raytrace.
`vie`	Displays a two-dimensional layout plot in Y/Z-plane.

4.2 Aspheric Surfaces Example

This example shows how aspheric surfaces can be used to improve the image performance of lenses. As a starting design, we restore a single lens of best form from the directory `examples\tutorial\BestformLens.otx`, as shown in Fig. 4.4. It exhibits a significant amount of spherical aberration, as can be seen in the fan aberration plot or the spot diagram (click on  or  to see the plots or type the commands '`fan`' or '`spo`' in the command line). Our aim is to reduce the spherical aberration by introducing aspheric deformation to the first surface.

Aspheric surfaces require the surface type ''A'', which replaces the default surface type ''S'', since surfaces can only be spherical (S) or aspheric (A).

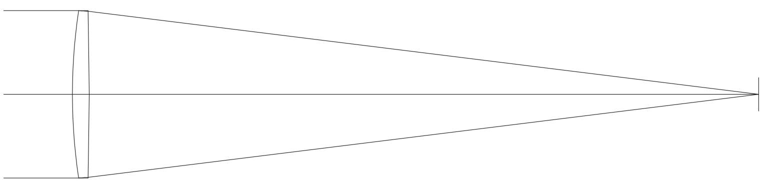


Figure 4.4: Best-form lens taken as staring example for aspherizing a lens.

4.2.1 Menu Entry

In the surface spreadsheet editor, we replace ''`S`'' by ''`A`'' in the surface type column on surface 1. Note that surface 1 is also the stop surface. It is therefore labelled '`STO`'.

For the sake of simplicity, we use conic sections only, these are parabolas, ellipses or hyperbolas. In the ''Asphere'' tab of the surface spreadsheet editor, the conic constant $K = -0.711$ must be entered in the column labelled ''K'' (the first column) on surface 1 (2^{nd} row), as shown in Fig. 4.5.

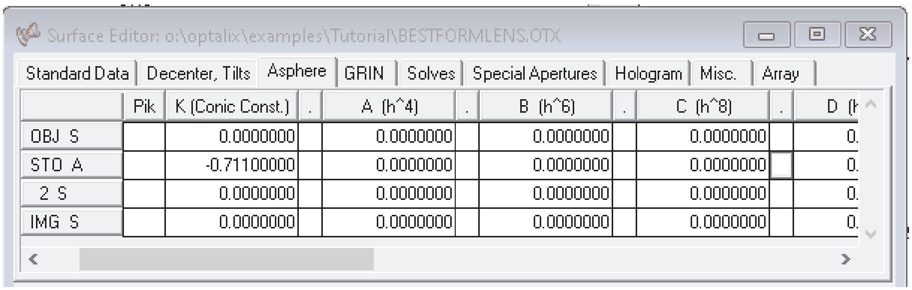
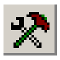


Figure 4.5: Entering conic constant K.

Notice that a value is only accepted in the spreadsheets if the Enter or Return key is pressed, or if the cursor is placed in another cell.

Repeating the spot and fan aberration plots reveals that the aberrations have reduced to almost unnoticeable levels. Change the plot scale for aberrations by clicking on the  icon (found to the left of each graphics window) and enter 0.001 (mm) in the subsequent dialog box to see the residual aberrations.

Higher order aspheric terms may be introduced to eliminate even this small spherical aberration. These coefficients, denoted by alphabetic characters A to F, are entered in the corresponding columns in surface spreadsheet editor as shown in Fig. 4.6. Enter the conic constant K and the first three coefficients, A through C :

K	-0.7333
A	0.143E-08
B	-0.232E-12
C	1.0E-21

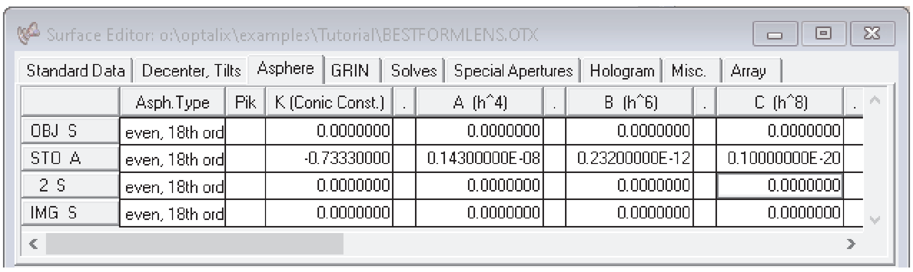


Figure 4.6: Entering higher order aspheric coefficients.

Even though additional coefficients (D to F) could be specified, there won't be any improvement in the correction of the spherical aberration.

4.2.2 Command Line Entry

In the command line, the staring design is restored by

`res c:\optix\examples\tutorial\BestformLens.otx`

if `c:\optalix` is the directory where *OpTaliX* is installed. The surface type (`SUT`) is then changed by

`sut s1 a`

The conic constant is specified by

`k s1 -0.7333`

Higher order aspheric terms are entered by

```
A s1 0.143E-08
B s1 -0.232E-12
C s1 1.0E-21
```

4.3 Gradient Index (GRIN) Example

Similarly to improving the on-axis performance of a single lens (by aspherizing a surface), we will use gradient index material to achieve a comparable image quality. Likewise we restore the single best-form lens example from `examples\tutorial\BestformLens.otx`. See Fig. 4.4 for the optical layout. Only two parameter need to be changed, the glass material and the so-called z-offset. The z-offset defines the point in the gradient index profile, where the material physically starts.

4.3.1 Menu Entry

In the spreadsheet, enter the Gradium^{TM} glass G51SFN on surface 1. The program recognizes that this type of glass is gradient index (GRIN) and automatically updates surface type to ''SI'' (S = **S**pherical, I = gradient **I**ndex) and the type of index profile to ''LPT''. This is an index of refraction profile defined by LightPath (LPT). The z-offset value 1.854 is entered in the ''GRIN'' tab in the Z-offset column on surface 1. Comparing the fan aberration plots of the homogeneous lens with the GRIN equivalent, a significant improvement could be achieved. However, even smaller aberrations can be achieved when the radii of the lens are fine-tuned in later optimization runs. Without diving into optimization details in this section, the interested reader is invited to enter the following parameters, which were obtained after a single optimization run:

Radius of surface 1 :	172.068240
Radius of surface 2 :	-889.854031
Z-offset :	5.89166
Defocus :	0.03670

4.3.2 Command Line Entry

The conversion of the single (best-form) lens to a gradient index (GRIN) lens is accomplished by the commands:

```
gla s1 g51sfn
gzo s1 1.854
```

The parameters to be changed for the lens after fine-tuning (optimization) are

```
gla s1 g51sfn
gzo s1 5.89166
rdy s1 172.068240
rdy s2 -889.854031
def 0.03670
```

4.4 Special Apertures Example

Unusual optical systems often require more complex aperture shapes other than those provided by the basic aperture shapes (circular, elliptical or rectangular). For this reason *OpTaliX* allows multiple (up to 10) basic apertures which can be logically combined on a surface.

Each basic aperture on an individual surface may be transmitting or obstructing, it can be decentered in X- and Y-direction from the local surface vertex and it can be rotated. Basic apertures may be logically combined by `.and.` respectively `.or.` operators. *OpTaliX* provides an operator `p` to address different basic apertures on a given surface.

A complex, yet easy to understand aperture example is given in Fig. 4.7. This is a typical aperture shape of compound two-mirror telescopes (Cassegrain or Ritchey-Chretien type) where the circular aperture (p1) of the primary mirror is obstructed by a central circular aperture (p2) of the secondary mirror. Additional obstructing apertures (`p3`, `p4`, `p5`) represent the spider arms holding the secondary mirror in place.

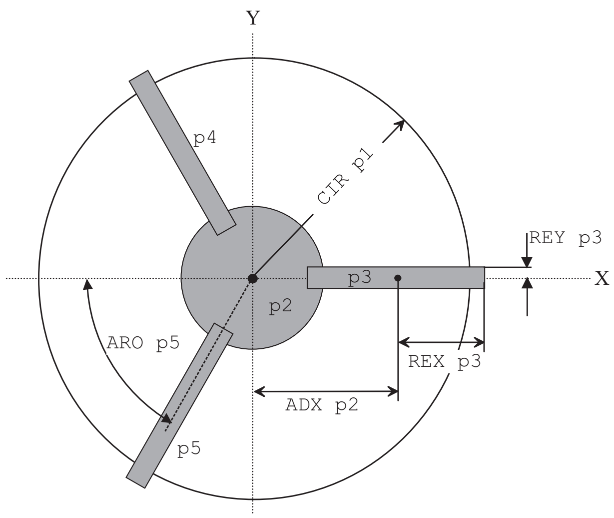
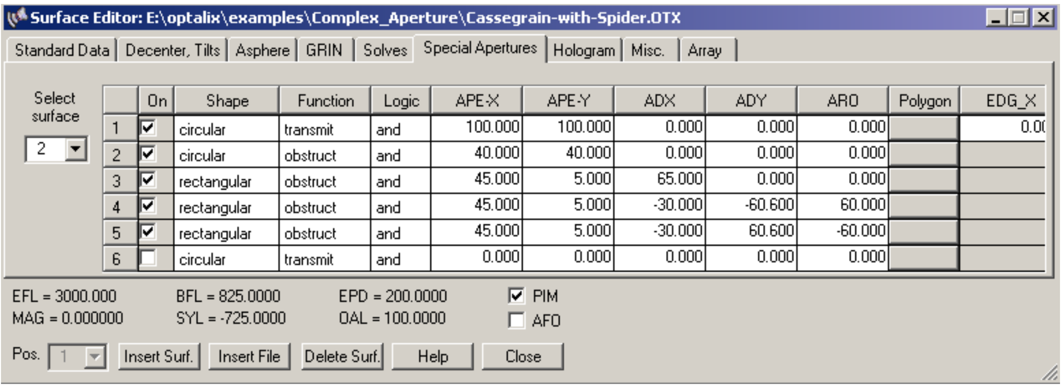


Figure 4.7: Circular aperture with circular central obstruction and spider with 3 vanes.

This aperture is modelled by 5 basic apertures which are logically `AND`ed. The following command sequence defines the basic apertures:

```
cir s1 100        ! defines circular transmitting aperture (p1).
!
cir s1 p2 40 obs  ! defines circular obstructing aperture (p2).
!
rex s1 p3 45 obs  ! rectangular obstructing aperture (p3).
rey s1 p3 5 obs
adx s1 p3 65
aro s1 p3 0
!
rex s1 p4 45 obs  ! rectangular obstructing aperture (p4).
rey s1 p4 5 obs
adx s1 p4 -30
ady s1 p4 -60.6
aro s1 p4 60
!
rex s1 p5 45 obs  ! rectangular obstructing aperture (p5).
rey s1 p5 5 obs
adx s1 p5 -30
ady s1 p5 60.6
aro s1 p5 -60
```

In the surface editor, special aperture tab, the aperture parameters are entered as shown in Fig. 4.8.



	On	Shape	Function	Logic	APE-X	APE-Y	ADX	ADY	ARO	Polygon	EDG_X
1	☑	circular	transmit	and	100.000	100.000	0.000	0.000	0.000		0.0(
2	☑	circular	obstruct	and	40.000	40.000	0.000	0.000	0.000		
3	☑	rectangular	obstruct	and	45.000	5.000	65.000	0.000	0.000		
4	☑	rectangular	obstruct	and	45.000	5.000	-30.000	-60.600	60.000		
5	☑	rectangular	obstruct	and	45.000	5.000	-30.000	60.600	-60.000		
6	☐	circular	transmit	and	0.000	0.000	0.000	0.000	0.000		

Figure 4.8: Surface spreadsheet editor: Defining a circular aperture with circular central obstruction and spider with 3 vanes.

The positioning of the aperture elements my be controlled by the ray intersection plot (RIS) at the surface of interest. This plot also shows the aperture borders. It is invoked from the command line by

```
spo ris sk scale
```

where sk is the surface number and scale is the maximum plot area. From the menu this plot is created by: *Geom.Analysis -> Ray Intersections*. See Fig. 4.9 for a sample plot.

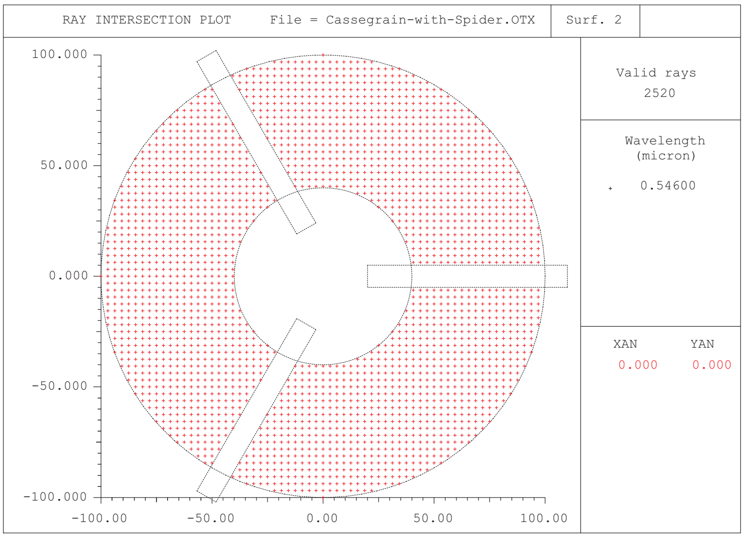


Figure 4.9: Ray intersection plot showing the used areas on a surface with central obstruction and spiders. The necessary apertures to produce this plot are shown in Fig. 4.8. This plot can be generated by restoring the file `$i\examples\complex_aperture \Cassegrain-with-spider.otx` and plotting ray intersections and aperture elements by the command `SPO RIS s2 100`.

4.5 Diffraction Grating Example

4.6 Solves Example

Solves allow the precise control of *paraxial* properties. Conditions for specifying a solve are, for example, holding the paraxial ray angle, the paraxial ray height or a certain paraxial ray incidence angle to a specified value. Solves will keep these requirements satisfied. For example, a paraxial ray angle solve at a surface will change its radius of curvature to maintain the specified ray angle. A paraxial height solve at a surface will change the axial separation to maintain a specified ray height at the subsequent surface. Please keep in mind that solves only apply to *paraxial* quantities.

In this exercise we want to keep the image plane exactly at the paraxial focus. This condition shall be hold even if other parameters (curvatures, thicknesses, ...) in the optical system change.

4.7 Zoom Example

A zoom lens system is a stereotype for a *multi-configuration* system, in which the optical elements are the same in all configurations, but the lens separations are different. *OpTaliX* does not only handle ''zoom'' systems in this classical sense but accepts virtually any construction parameter to build variable systems. Based on this capability *OpTaliX* can handle applications such as

- classical zooms (changing lens separations only),
- scanning systems (changing surface tilts and/or decenter),

- multi-spectral systems (changing wavelengths or complete spectra),
- multiple eye positions for display systems,
- multiple conjugate systems (macro lenses),
- flip in/out attachments, etc.

The word ''zoom'' will be used throughout this tutorial to describe systems that change configuration. The number of zoom positions is unlimited, however, a maximum of 100 positions can be edited in the zoom spreadsheet editor[1]. All zoom data are saved as part of the lens. Zoom/multi-configuration lenses may be ''de-zoomed'' to any selected zoom position.

Usually the starting point is a non-zoomed lens[2].

In this section we restore a classical Zoom lens design found in the examples directory `\examples \zoom_Multiconfigurtion \Laikin-35-1.otx`. This is a 10x mechanically compensated lens taken from Laikin's book ''Lens Design'' [29]. The optical layout and the movements of the lens groups is shown in Fig. 4.10.

The zoom lens consists of four groups constituted by the surface ranges 1-5, 6-10, 11-15, and 16-25. The outer groups (surfaces 1-5 and 16-25) are stationary while the two inner groups (surfaces 6-10 and 11-15) are moved in opposite direction to each other. The air spaces 5, 10, 15 (measured right to the surfaces) define the movement of the lens groups, whereas the sum of the thicknesses (thi s5 + thi s10 + thi s15) must be constant. This way the focal length (EFL) is changed by a factor 10 from approximately 5.8mm to 58mm.

Since the continuous change of focal length is modelled at discrete points (zoom positions), we must allow different values of the thicknesses. That is, thicknesses that are floating must be specified separately at each discrete position.

In the command line this is accomplished by the ZOO prefix to each command. For example, the thickness `thi s5` becomes, depending on the zoom position:

```
zoo thi s5 1.33 24.35 40.13 50.95
```

which may also be expressed by individual statements

```
thi s5 z1  1.33
thi s5 z2  24.35
thi s5 z3  40.13
thi s5 z4  50.95
```

Notice the usage of the zoom qualifiers (`z1 to z4`) as explained in sect. 3.1.4. Similarly thickness of s10 and s15 are defined. Fig. 4.11 shows the zoom setup in the zoom editor (invoked by the command `edi zoo`, or by clicking on the icon 

).

[1] For more than 100 zoom/multi-configuration positions a text based editor is provided. It is invoked by the ZED command.

[2] We will also use the term ''fixed'' for non-zoomed systems, because all construction parameters are constant (fixed). In particular, this terminology applies to the optimization of zoom systems as given in section 6.2.

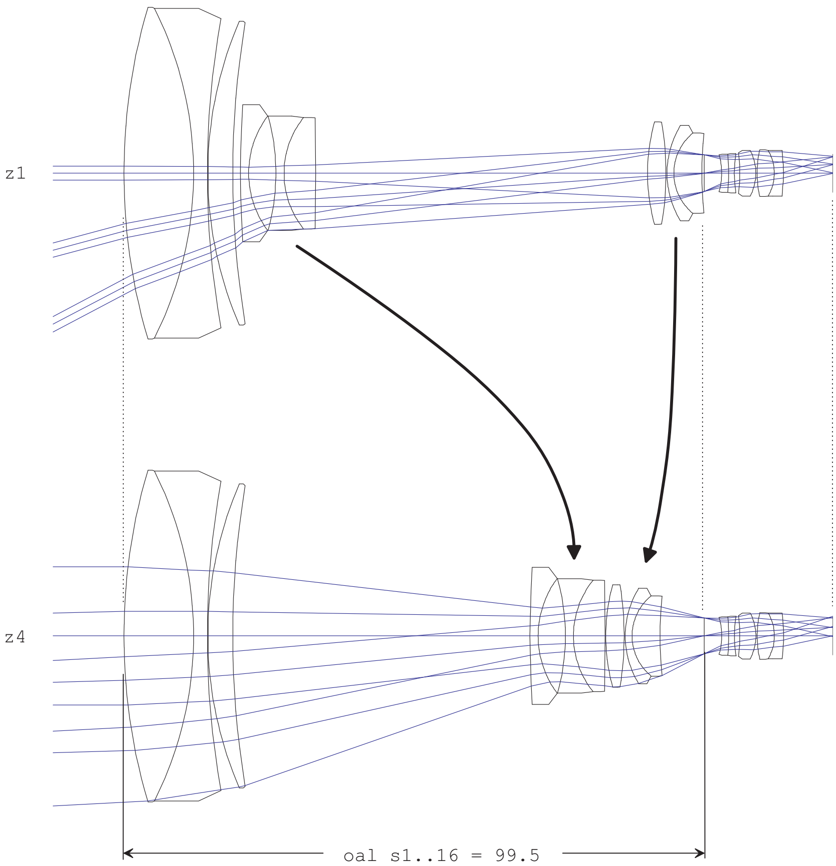
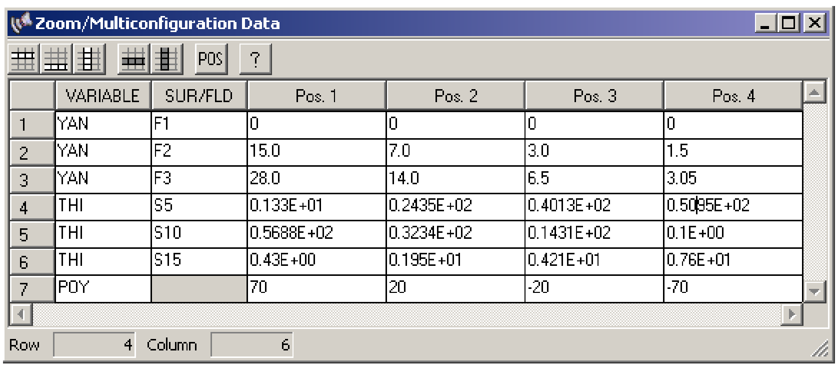


Figure 4.10: Mechanically compensated zoom lens from Laikin [29], example 35-1. Here shown for zoom positions z1 and z4. The middle positions (z2 and z3) have been left out for sake of clarity.



	VARIABLE	SUR/FLD	Pos. 1	Pos. 2	Pos. 3	Pos. 4
1	YAN	F1	0	0	0	0
2	YAN	F2	15.0	7.0	3.0	1.5
3	YAN	F3	28.0	14.0	6.5	3.05
4	THI	S5	0.133E+01	0.2435E+02	0.4013E+02	0.5095E+02
5	THI	S10	0.5688E+02	0.3234E+02	0.1431E+02	0.1E+00
6	THI	S15	0.43E+00	0.195E+01	0.421E+01	0.76E+01
7	POY		70	20	-20	-70

Figure 4.11: Zoom editor

The syntax for defining zoom definitions is

```
zoo variable SFW param_pos1 param_pos2 ... param_pos_n
```

where SFW is a place holder for either sk, fk, wk, pk, which are surface, field, wavelength and pupil qualifiers. '`variable`' is any construction parameter that can be edited in the command line. The number of parameters should match the number of zoom positions defined. For example, zooming a field point on the command line,

```
zoo yan f2 15.0 7.0 3.0 1.5
```

assigns different Y-coordinates (`yan`) to field 2, depending on the zoom position. Note the `zoo` prefix in the command line which is obligatory to distinguish a zoom command from a normal (fixed position) command. The `zoo` prefix is not required in the zoom editor.

The `lis` command lists the complete zoom definition:

```
Zoom Data :
 ZOO YAN   F1    0              0              0              0
 ZOO YAN   F2    15.0           7.0            3.0            1.5
 ZOO YAN   F3    28.0           14.0           6.5            3.05
 ZOO THI   S5    0.133E+01      0.2435E+02     0.4013E+02     0.5095E+02
 ZOO THI   S10   0.5688E+02     0.3234E+02     0.1431E+02     0.1E+00
 ZOO THI   S15   0.43E+00       0.195E+01      0.421E+01      0.76E+01
 ZOO POY         70             20             -20            -70
```

4.8 Global Surface References Example

Global referencing is a method to refer the local coordinate system of a surface to the coordinate system of a previous surface. Sometimes this method is denoted as a RETURN to the coordinate system of a prior surface or as a '*coordinate return*' .

Global referencing breaks the strict sequential order of surfaces (where the local coordinate system of a surface refers to its immediately preceding surface), even though the ray trace remains sequential.

Global references offer great flexibility as they allow positioning of surfaces independent of the position of prior surfaces, reduction of variables in optimization, and allows building of complex opto-mechanical relationships. This feature is particularly interesting in systems containing tilted/decentered surfaces and in scan- or zoom-systems.

A sensible application of global references is the zoom example given in the previous section 4.7. See also Fig. 4.10 for the layout plot. The first and fourth lens group are coupled together and must maintain a constant separation while the two inner groups move in axial direction to accomplish the focal length change. We have seen that three air spaces (thicknesses 5, 10 and 15) need to be defined, as the model is strictly sequential and the position of each surface is defined in relation to the position of its prior surface.

By referring the axial position of the fourth lens group (surfaces 16-24) to any arbitrary surface in the first lens group (which is stable), we can eliminate one thickness parameter (thi s15). That is, if the position of the fourth lens group is defined at the coordinate system of the first group (preferably surface 1), we do not need to worry about thickness 15. The program automatically calculates it. Fig. 4.12 indicates this condition.

Global references are defined by the `glb` command. To define surface 16 in the coordinate system of surface 1 we write in the command line

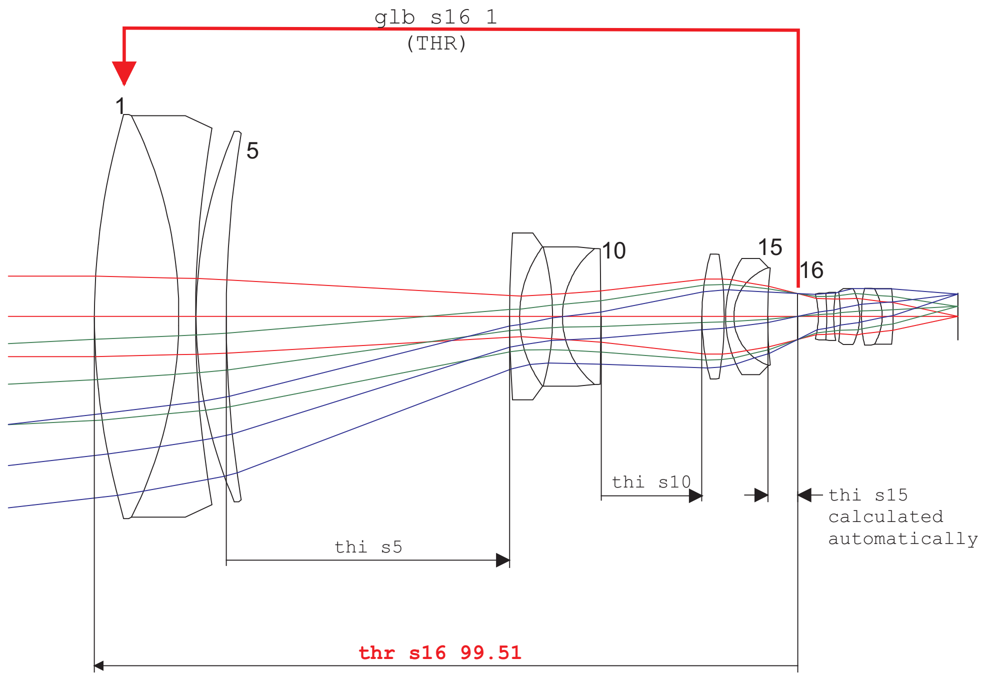


Figure 4.12: Global reference. In this example, the coordinates of surface 16 are referred to surface 1. The reference thickness (THR) is the axial thickness measured 'left' towards the reference surface.

```
glb s16 1
```

When we execute the glb s16 1 command, the position of the 16th surface is automatically placed with respect to the 1^{st} surface. The program automatically calculates the distance of surface 16 from surface 1. This value, denoted by `THR` is 99.51mm. THR can be changed to any value by the user. Since the position of surface 16 is now independent of the preceding surface, the thickness of surface 15 in front of surface 16 has become a dependent variable and is automatically calculated. Therefore, THI s15 cannot be entered by the user.

Surface 16 will be set to Tilt Mode: NAX (new axis) which defines a new new reference coordinate system.

Alternatively, in the surface editor, the global reference is entered in the "Glb" column in the 'General' tab (see Fig. 4.13). The axial position of surface 16 (with respect to surface 1) is determined by the reference thickness (`THR`) parameter. This is the axial thickness to a referenced surface. It is automatically calculated on redefining a global surface reference. In addition, the zoom parameter (`zoo thi s15 ...`) is no longer needed and can be deleted.

In this context we need to explain the concept of the reference thickness (`THR` command). Normally axial thicknesses are defined 'right' to the surface, i.e. it specifies the location on the Z axis of the next surface relative to that of the current surface. If we assume a surface k which is globally referenced to a prior surface, then the axial position of surface k is defined by the reference thickness `THR` which is always measured 'left' to surface k. Thus, a globally referenced (`GLB`) surface k possesses two thicknesses: A thickness (`THI`) defining the axial distance to the subsequent surface (the usual sequential mode) and a reference thickness (`THR`) defining the axial position of surface

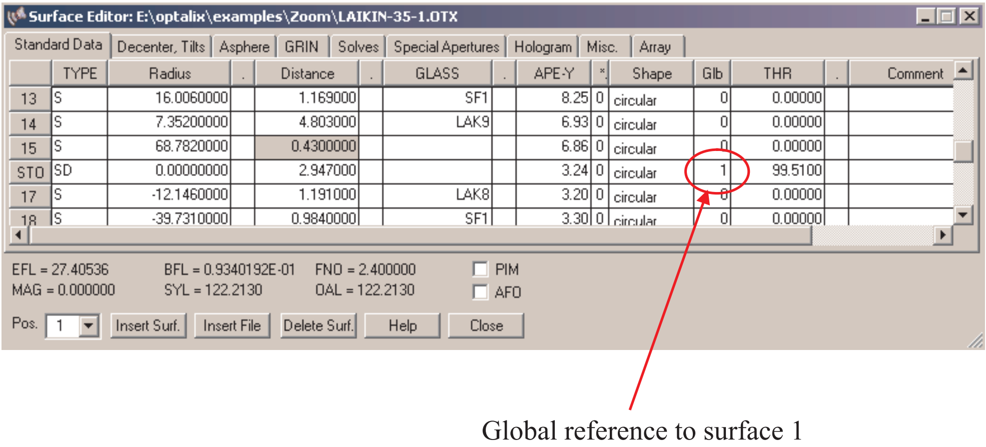


Figure 4.13: Defining global reference in the surface editor. In this example, the coordinates of surface 16 are referred to surface 1.

k with respect to a prior surface.

5

Special Options

5.1 Macros

5.2 User Defined Graphics (UDG)

This page is blank.

6

Optimization

Optimization is the process of minimizing aberrations by changing the values of some of the lens data (variables) to meet certain requirements in a given application. In this section, we will explain the concepts used in $OpTaliX$ and all the necessary steps to meet this goal. A more detailed description of the optimization algorithms is found in the reference manual.

A successful optimization setup requires the definition of both variables and a relative error function. The variables are typically quantities such as surface curvatures, thicknesses, tilt angles, etc. The performance of an optical system is measured by a user-defined error function, the so-called merit function. The merit function gives an estimate of the difference in performance related to the design requirements. The lower the merit function is, the better is the performance.

Optimization is an iterative process in which the variables are changed repeatedly in the attempt to find new values for the variables that yield a lower merit function. Therefore, the first and most basic step is a starting design, which is either derived from first order principles or taken from the literature. The examples library delivered with $OpTaliX$ may serve as a good reference in finding the starting point.

The optimization algorithms used in $OpTaliX$ are searching for the *local minimum* of a given function and therefore the achievable level of aberration correction depends on the choice of the starting point. This is schematically indicated in Fig. 6.1 for one variable.

The number of variables, however, is typically much greater than one and the number of local minima is typically very large. This makes the choice of the starting point crucial. Although the initial starting points A and B in Fig. 6.1 have about the same merit function, optimization will yield different minima of the merit function. It should be noted that finding the *global minimum* (or a suitably low minimum) is up to the designer's skill and experience.

We will now describe the steps for setting up all required optimization parameters. These are

a) definition of variables
b) definition of targets and constraints, which form the merit function
c) running the optimization

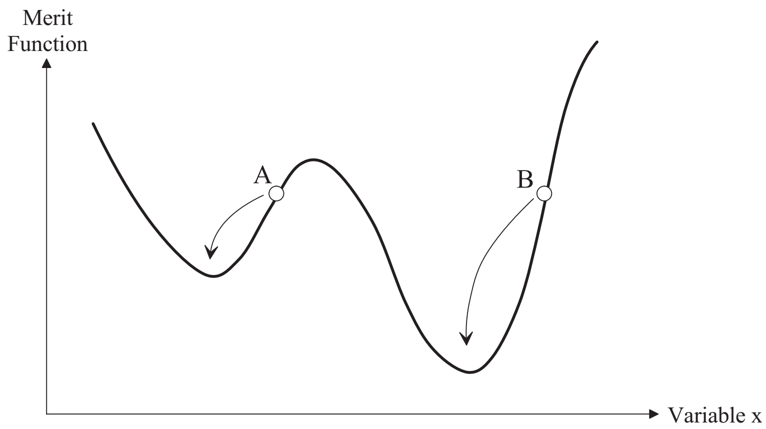


Figure 6.1: Merit function for one variable. The choice of the starting point (A, B) determines which local minimum is found.

6.1 Fix-Focus Lens Optimization

In this section we will optimize a simple apochromatic telescope objective, which has been originally proposed by Buchroeder. The design is found in the examples library under `\examples\optimization\apo_0.otx`. A plot of the transverse ray fan (use `FAN` command or click on the icon) shows its poor correction status, because Buchroeder's design has been artificially degraded for the purpose of our optimization exercise.

6.1.1 Defining Variables

Construction parameters are made variable by the command "`EDI VAR`" or by clicking on the OPT VAR icon. A dialog box as shown in Fig. 6.2 will be opened. In our Buchroeder design, we will make all curvatures variable, that is `CUY` (curvature Y) at surfaces 1 to 4. Thus, we check surfaces 1 to 4 in the CUY column.

In addition, we will also make the defocus (DEF) variable. The defocus is an additive value to the paraxial focus and is entered as the thickness of the last surface, the image surface. Since the thickness after the image has no physical meaning it is used for the defocus. The defocus may be entered in the surface editor as the thickness of the last surface, its value is made variable by checking the `THI` field at the image surface in Fig. 6.2.

Use of the defocus parameter helps finding the minimum spot diameter, because the optimum spot is generally not at the paraxial focus, but at a small offset (the defocus), mainly due to residual spherical aberration.

6.1.2 The Merit Function

The merit function is composed of targets and constraints. Targets are, for example, a minimum spot diameter (SPD) or a minimum lateral chromatic aberration (LAC). A constraint is a parameter,

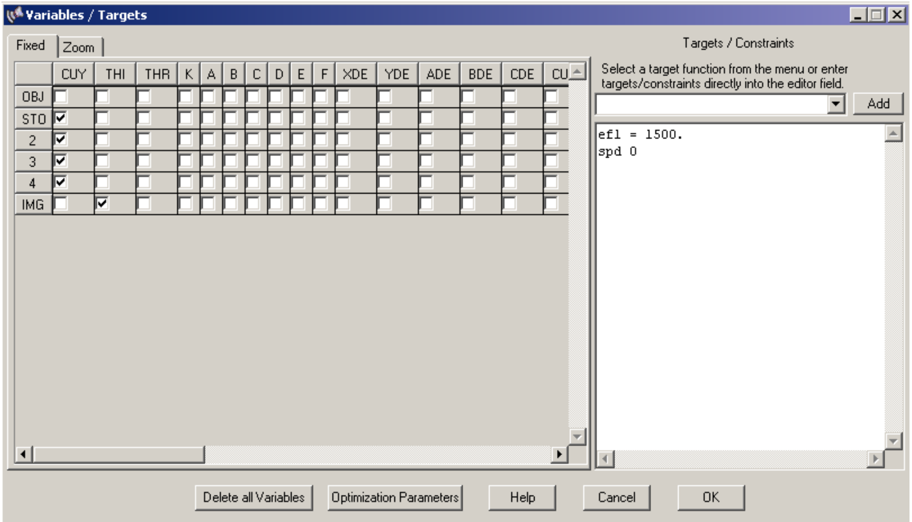


Figure 6.2: Dialog box for defining variables, targets and constraints in the Buchroeder apochromatic refractor example.

which shall be hold exactly or shall be greater or smaller than a specified value. For example, holding the focal length (EFL) to a precise value is a constraint.

The construction of the merit function in the Buchroeder example is quite simple. In the editor field of the Variables/Targets dialog box, we enter

```
EFL = 1500
SPD 0
```

The meaning should be obvious: We hold the focal length (EFL) to exactly 1500 mm. The target for the spot diameter (`SPD`) is zero. Since no operator ($<, >, =$) is specified in `SPD`, the spot diameter will be minimized for *all* fields and wavelengths defined in the optical system. This is merely a minimum merit function definition, however, it is sufficient for our first example. We will learn later on how to construct more advanced merit functions.

Targets, respectively constraints, may also be selected from the list box in the upper right corner of the dialog box as shown in Fig. 6.3. Pressing the ''Add'' button to the right of the list box adds the target function to the merit function definition. Note that only the function name is added and that the corresponding entry to the merit function must be completed by adding an appropriate operator ($>, <, =$) and a target value (or constraint). In the example below (Fig. 6.3, only the target function

```
EFL =<> target
```

is added as a new entry to the merit function. The tokens `=<>` and `target` are placeholders which should be appropriately edited. For example,

```
EFL = 1500
```

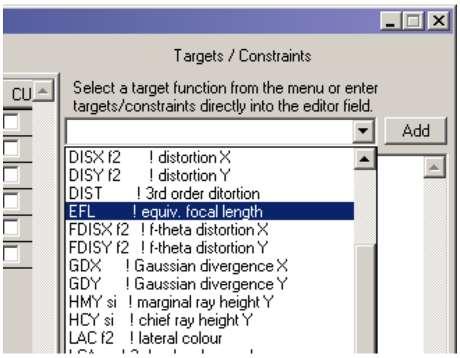


Figure 6.3: Selecting targets/constraint functions from a list box and adding it to the merit function definition.

Having finished the definition of the merit function, we close the dialog box.

At this point it is worthwhile to note that the merit function may consist of almost any *OpTaliX* command. For example, entering `SPD` at the command line would print the rms-spot diameter in the text window. The same command, but now used in the definition of the merit function, minimizes the spot diameter. However, entries to the merit function may be quite complex as arithmetic expressions (such as `2*sqrt(2)/3`) and lens database items (such as thickness, radius of curvature, etc.) may also be used to construct targets. Thus, it is perfectly valid to write the constraint on focal length as

```
EFL = 2*750
```

or

```
EFL = sqrt(1500)**2
```

See the reference manual for a list of commands/expressions to be used in optimization. All required optimization parameters are now defined and we are ready to run the optimization.

6.1.3 Run the Optimization

An optimization run is performed by the command OPT at the command line or by clicking on the OPT icon. By default, 10 iteration steps are done. Basically, the output from the optimization run can be divided into four sections, as shown below:

Part 1: Number of variables and constraints:

```
KT Optimization:
Number of variables            :    5
Number of functions            :  900
Number of equality constraints :    1
Number of inequality constraints :  0
```

Part 2: The error function components at optimization start:

```
Targets/Constraints                    Target       Function         Error
efl = 1500.                      1500.000000    1457.939417    -42.060583
spd 0                               0.000000       0.038304      0.038304
```

Part 3: Error function report for each iteration:

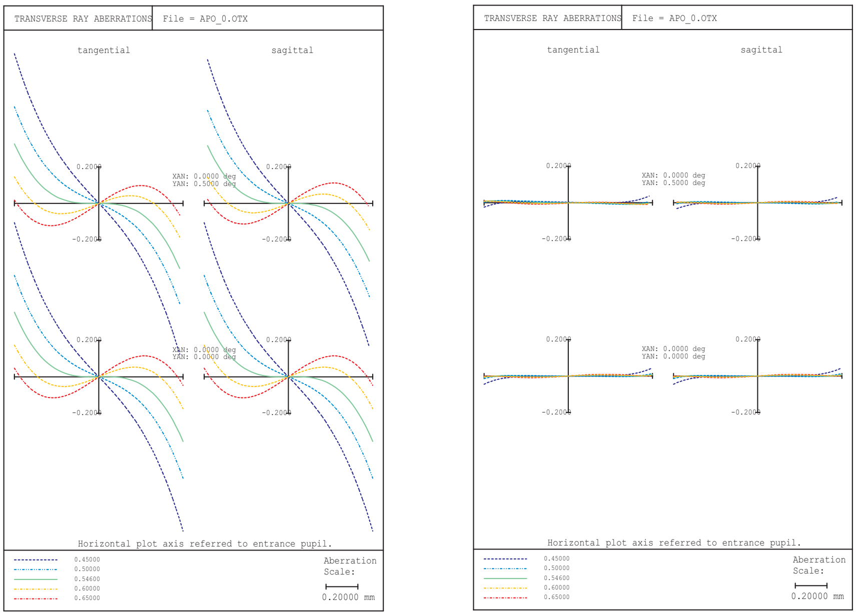
Iter	Min.	Equal.	Inequal.	DumpingF.	Improv.
0	0.906432	6.485413	0.000000	1.000000	
1	0.268796	1.105826	0.000000	0.8133397E-02	0.70346
2	0.041503	0.102830	0.000000	0.8133397E-03	0.84560
3	0.027474	0.157272	0.000000	0.8133397E-07	0.33801
4	0.027454	0.003427	0.000000	0.8133397E-13	0.00073

```
Optimization stopped. Improvement is less than    0.01000 (1.00%)
```

Part 4: The error function components after optimization has terminated:

Targets/Constraints	Target	Function	Error
efl = 1500.	1500.000000	1500.000012	0.000012
spd 0	0.000000	0.001160	0.001160

A comparison of the ray fans as shown in Fig. 6.4 demonstrates the improvement of correction.



before optimization after optimization

Figure 6.4: Transverse ray aberration (fan curves) of the Buchroeder design before and after optimization.

6.1.4 Example 2

In this example, we will optimize a simple eyepiece of the Ploessl type. The optical layout of the starting design is given in Fig. 6.5. This example is found in the examples library under `\examples\optimization\eyepiece_0.otx`. All optimization parameters as discussed below are already set after the optical system has been restored. As in the previous section, the aberrations are intentionally very large. The goal of this exercise is to keep the beam at the image

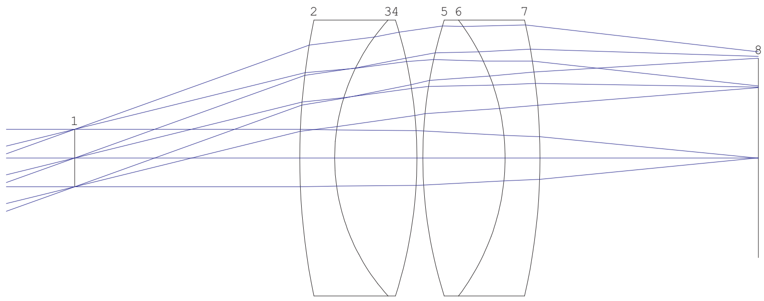


Figure 6.5: Ploessl eyepiece.

plane telecentric, that is the principal rays are parallel to the optical axis in image space.

In order to force telecentricity, we need a constraint on the direction of the chief rays. This is accomplished by the `CY` command. `CY` is the direction cosine of a ray. Quite a bunch of additional parameters are required to defining exactly one single ray. The full constraint definition would be

```
CY s8 f2 w1 0 0 = 0
```

This looks complicated at first sight, however, the constraint is easily defined if the underlying concept is understood. This constraint (command) specifies the direction cosine in Y-direction (`CY`) on the 8^{th} surface (s8), field number 2 (f2) and wavelength number 1 (w1). The ray goes through the stop at the relative coordinates x = 0, y = 0, this is the center of the stop. The equal operand (=) specifies that this condition shall be fulfilled exactly with a target value 0.

In addition we want to maintain focal length at 25mm and want the spot diameter for all field points minimized. The complete merit function definition is then

```
efl = 25
CY s8 f2 w1 0 0 = 0
spd 0
```

The variables are curvatures (`CUY`) at surfaces 2 to 7 and the defocus term (`DEF`). Fig. 6.6 summarizes all required entries in the variables/constraints dialog box (invoked by the `EDI VAR` command or by clicking on the OPT VAR icon).

Performing an optimization run yields a significant improvement of the off-axis aberrations as shown in the before-after comparison in Fig. 6.7. It is interesting to know how good the telecentricity constraint has been fulfilled. In the command line, we query the direction cosine CY using the evaluate (EVA) command:

```
eva [CY s8 f2 w1 0 0]
```

The result yields a direction cosine `-6.279830604638548E-06`, which is approximately -0.00036

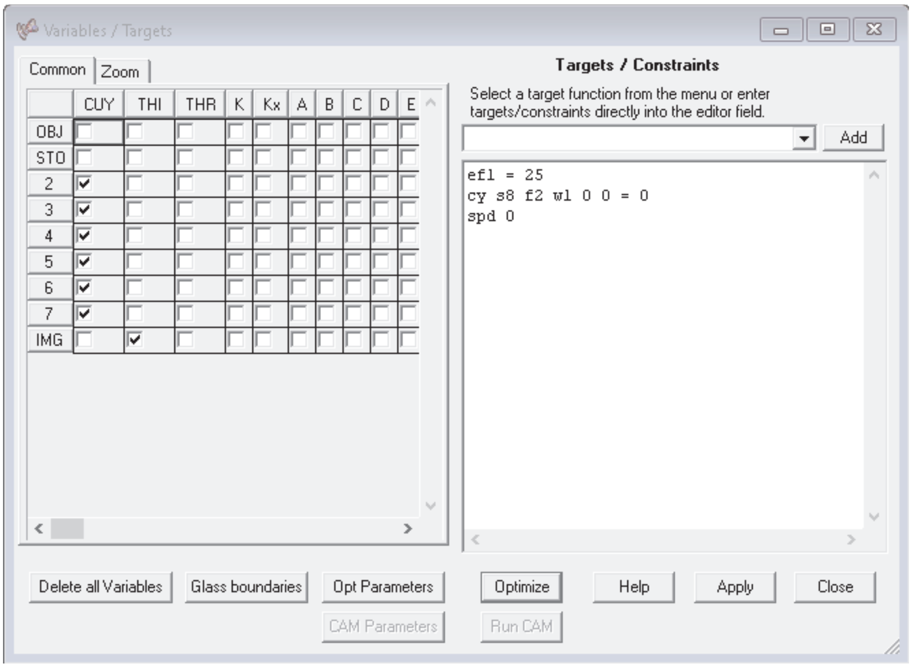
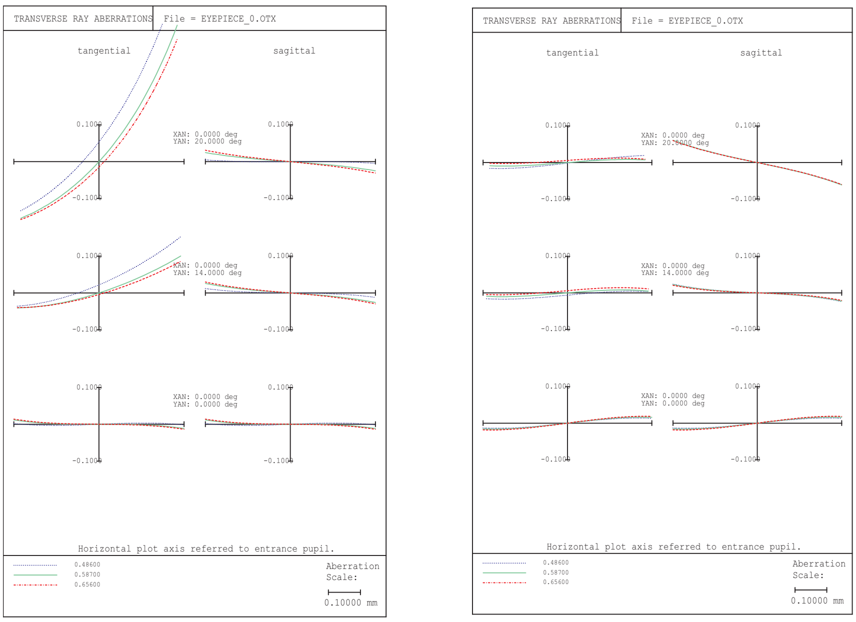


Figure 6.6: Dialog box for defining variables, targets and constraints for the Ploessl eyepiece.

degree ($\sin^{-1}$-6.279830604638548E-06). For most practical purposes this value is sufficient.



before optimization

after optimization

Figure 6.7: Transverse ray aberration (fan curves) of the Ploessl eyepiece design before and after optimization.

6.1.5 Setting General Optimization Parameters

General optimization parameters control the behaviour of the optimization (for example selecting an appropriate optimization method) and define which values are reported during optimization.

This section provides complementary information for adjusting the optimization process. Since we have already predefined reasonable optimization parameters in all optimization examples found in this tutorial there is no immediate need to alter these values. The reader may skip this section and return to it at a later time.

General optimization parameters are defined in a dialog box accessed from the main menu *Optimization −− > Parameters*.

6.1.5.1 General Tab

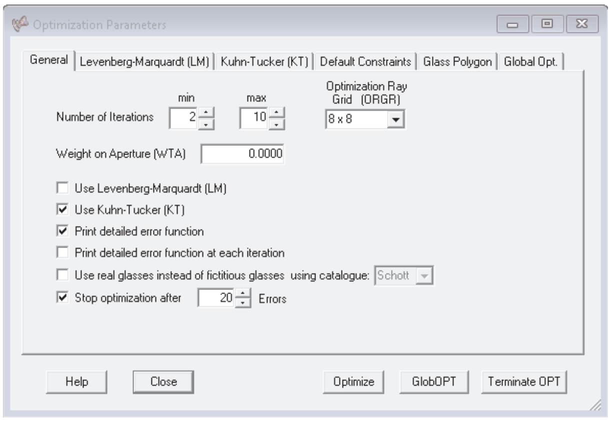


Figure 6.8: Dialog box for editing general optimization parameters.

A minimum and maximum number of iterations can be set. Optimization will cycle at least the minimum number of iterations and will not terminate earlier, irrespective whether the system improves or not.

Optimization will be terminated if the maximum number of permitted cycles are completed. Termination will probably occur before if the fractional improvement is less than the improvement factor (see `IMPR` command).

OpTaliX permits a different ray grid in optimization than the ray grid normally used in performance analyses. The optimization ray grid defines the number of rays across the pupil diameter only during optimization. This setting must not be confused with the number of rays used for analyses (see NRD command). Setting the optimization ray grid to a lower value than NRD will only reduce the number of rays during optimization. For example, selecting an optimization ray grid 16x16 and NRD 32 will only use every second ray in the ray matrix during optimization. This accelerates the speed of optimization by a factor 4, whereas all analyses performed subsequently will still use the 32x32 ray grid.

As for the optimization algorithm to use, *OpTaliX* currently offers two different **local** optimization methods, the Levenberg-Marquardt (LM) optimization and the Kuhn-Tucker (KT) optimization, plus a global optimization (GO) based on the so-called ”escape” function proposed by M.Isshiki.

The global optimization, however, will not be discussed here as it is an advanced subject for very experienced users. Consult the reference manual for a detailed description of the optimization methods.

By default, the KT-optimization is used. This setting should not be changed unless otherwise recommended.

Print detailed error function: If checked, the components contributing to the error function and the total error function value will be printed at optimization start and after optimization has finished.

Print detailed error function at each iteration: As the description of this check box implies, error function components and total error function value will be reported between each iteration step. Note that this option may increase output in the text window significantly and should only be activated for diagnostic reasons.

6.1.5.2 Levenberg-Marquardt (LM) Tab

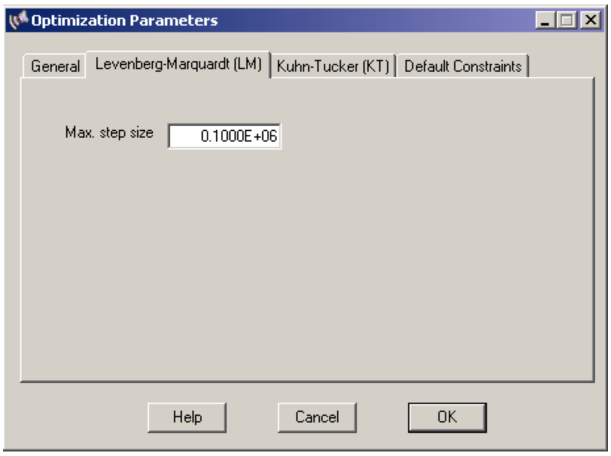


Figure 6.9: Dialog box for editing options in the Levenberg-Marquardt (LM) optimization method.

This tab only provides a single parameter, the maximum step size. The value may be increased if convergence is slow or decreased if convergence is instable. The maximum step size need not normally be altered for most applications.

6.1.5.3 Kuhn-Tucker (KT) Tab

Show active constraints for each cycle: Outputs which constraints are currently active. Inequality constraints are dynamically activated or released as the optimization evolves.

Show variables sensitivity: If checked, reports the relative contribution of each variable parameter to the change of the merit function.

Fractional improvement: Optimization is terminated if the improvement of the error function is less than a certain improvement. Termination will probably occur before if the maximum number of cycles is exceeded.

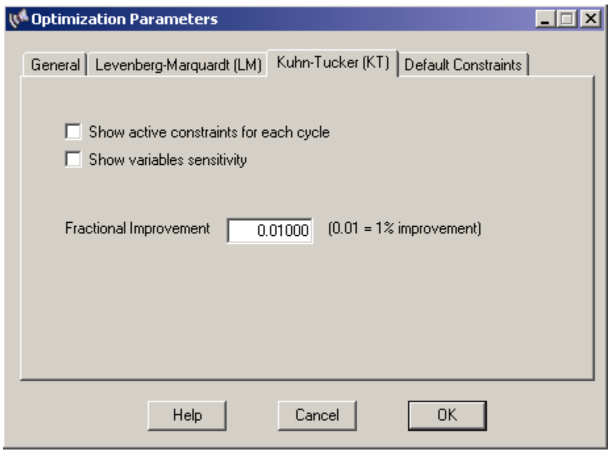


Figure 6.10: Dialog box for editing options in the Kuhn-Tucker (KT) optimization method.

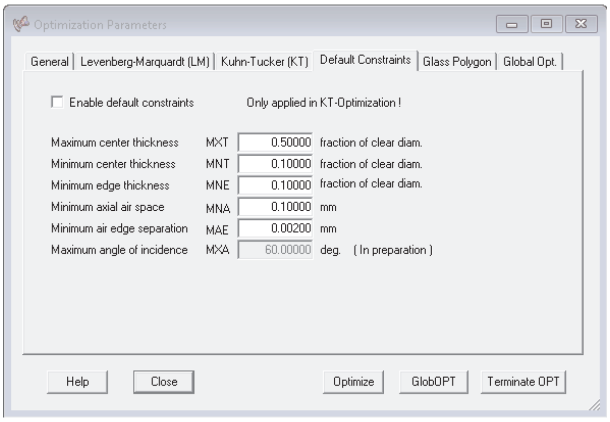


Figure 6.11: Dialog box for editing default constraints.

6.1.5.4 Default Constraints Tab

If enabled, default constraints will automatically be added to the list of target (error) functions. Default constraints are useful for maintaining reasonable dimensions of lenses and air spaces during optimization. For example, default constraints ensure that edge thicknesses are always manufacturable (i.e. greater than a certain fraction of the lens diameter) and that lenses do not intersect (i.e. air edge separation is always positive).

6.2 Zoom Lens Optimization

The reader should be familiar with the conventions for setting up zoom(ed) lens systems as described in section 4.7 (page 43). Optimization of zoom (or multi-configuration) lenses is more complex than a fix-focus lens as usually more variables and constraints need to be defined. This section gives a detailed procedure for setting up all required parameters. The design to be optimized is found in the examples library under `\examples\optimization\zoom-lens_o.otx`. It is an artificially degraded zoom lens from Laikin, ''Lens Design'', 3^{rd}Ed., example 35-1. The optical layout and the movement of the groups are shown for two positions in Fig. 6.12.

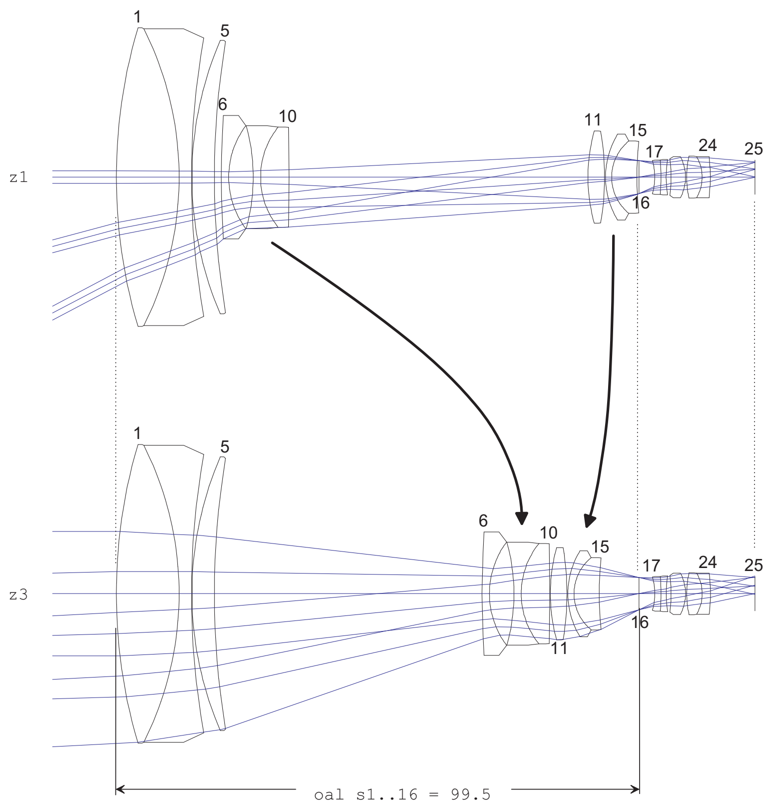


Figure 6.12: Mechanically compensated zoom lens shown for two positions, z1, z3. The middle position z2 has been left out for the sake of clarity.

The zoom lens consists of four groups constituted by the surface ranges 1-5, 6-10, 11-15, 16-25. The outer groups (surfaces 1-5 and 16-25) are stationary while the two inner groups (surfaces 6-10 and 11-15) are moved in opposite direction to each other, as indicated by the curved arrows in Fig. 6.12. This way the focal length (EFL) is changed by a factor 10 from approximately 5.8mm to approximately 58mm.

As can be seen in Fig. 6.13, the optical performance of our starting system is poor because its construction parameter have been slightly altered from the optimum for educational purposes only.

We will now define the variables. In this process we must distinguish between parameters which are identical for all zoom positions and those parameters which will take on separate values in each position. We will denote the first kind of variables as ''fixed'' variables[1] (or non-zoomed variables)

[1] we are aware that this terming is an oxymoron, however, 'fixed' in this context applies to variables in a fixed (non-

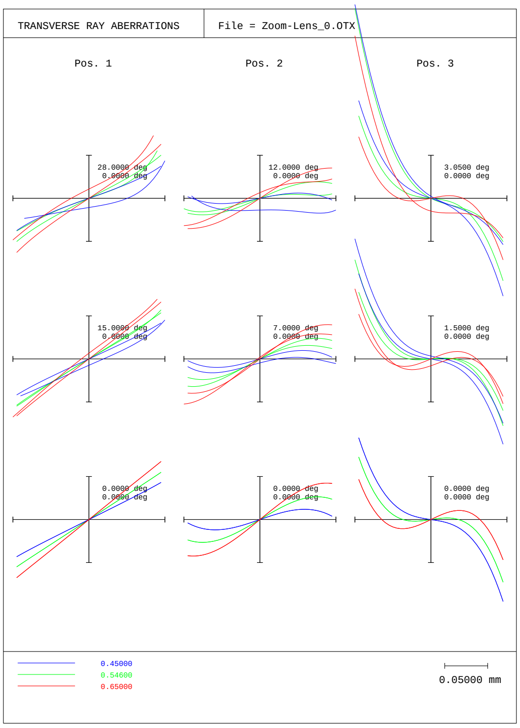


Figure 6.13: Transverse aberrations of the start system (before optimization). Note: If the appearance of your transverse aberration plot is different from this figure, one of your settings in the preferences section will be different. From the main menu *File −− > Preferences*, select the 'operations' tab and check 'Align ray fan curves horizontally'.

and the latter as ''zoom(ed)'' variables. Clearly the radii of curvatures must remain the same for all positions. Among the thicknesses, three air spaces will vary for each position. These are `THI s5`, `THI s10` and `THI s15`, which control the axial positions of the inner lens groups.

Definition of variables is accomplished from the toolbar by clicking on the OPT VAR icon or by entering the command "`EDI VAR`". In the following dialog (see Fig. 6.14), to the left, a matrix of check boxes is shown and two tab controls over there. By default, the left tab control ''Fixed'' is selected. We check all boxes in the leftmost column labelled ''CUY'' (curvature) except for surfaces 16 (stop) and 25 (image). Now, all curvatures are marked variable.

Click on the tab ''Zoom'' and a new check box matrix will appear with all boxes initially unchecked. Here we check all distances which are allowed to vary for each zoom position. As already indicated above, these are the thicknesses 5, 10 and 15 in the column labelled ''THI''. The result is shown in the right part of Fig. 6.14.

At this point it is worth to note why a distinction between zoomed and non-zoomed parameters must be made. A parameter which is made variable in the zoom control will generate N variables, where N is the number of zoom positions, whereas a non-zoomed (fixed) parameter only creates one single variable.

We will now define optimization constraints, that is, which aberrations shall be minimized and which operating conditions shall be maintained. First of all we define the focal length (EFL) for

zoomed) system.

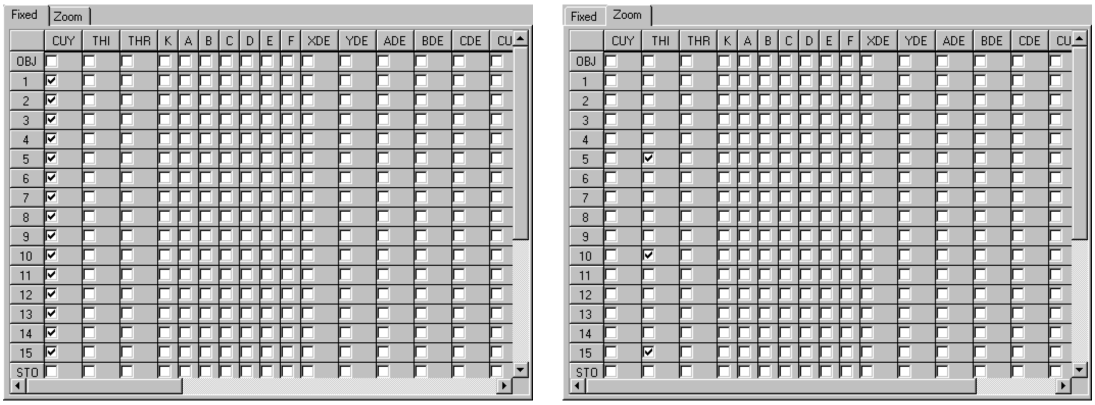


Figure 6.14: Editing variables. Left: Tab control for editing ''fixed'' (non-zoomed) parameter, right: tab control for editing ''zoomed'' parameter.

each zoom position. Since we have defined three zoom positions, three focal length constraints must be entered in the text box to the right of the variables/targets dialog box (use VAR command to open the dialog box again if you have closed it). The entries in the targets/constraints settings are given in Fig. 6.15.

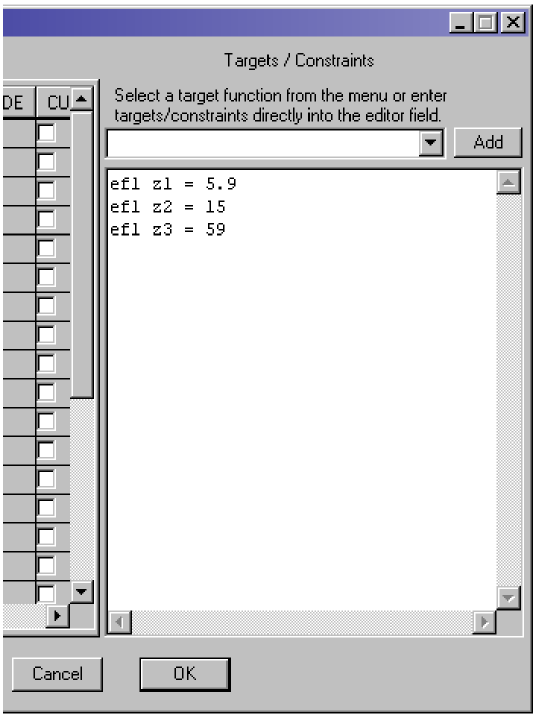


Figure 6.15: Defining focal length constraints for three zoom positions.

Of course, one of the main goals of optimizing a lens system is to achieving a good image quality, thus, we add criteria to minimize the spot diameter. The corresponding command for spot diameter in *OpTaliX* is SPD. We add the following lines to the constraints definition

```
spd f1..2 0
```

```
spd f3 0 ; wt = 0.7
```

which defines a zero target for spot diameter on fields 1 to 2 and a separate spot diameter target on field 3, however, with a lower weight (0.7) because we allow the performance at the rim of the field of view (field number f3) to be slightly worse than in the center and at half field of view. Make sure to separate all qualifiers by at least one blank character. The constraints/targets settings now should look identical to Fig. 6.16.

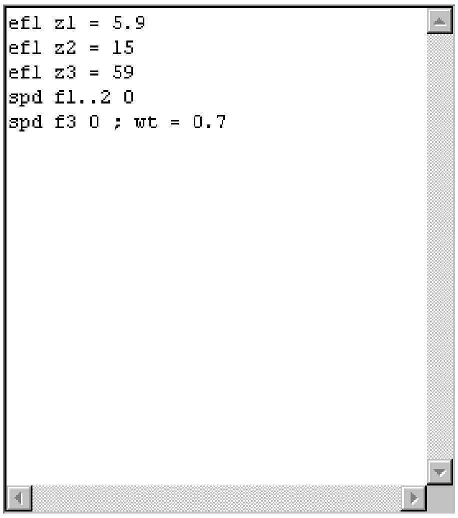


Figure 6.16: Adding spot diameter minimization to constraints/targets definition.

We are now at a point where optimization already could be run, however, the result would be unsatisfactory. Typically a zoom lens should maintain focus during ''zooming'' and we also need some additional constraints to keep the first and last group stationary. The latter requirement is fulfilled by requiring a constant over all length (OAL) between the first and last group and we add

```
oal s1..16 = 99.5
```

to the constraints definition. Compare also with Fig. 6.12 which indicates this thickness constraint. Note the absence of any zoom position qualifier in the definition above, which implies *all* positions. Lastly we require non-negative thicknesses between the two inner (movable) zoom groups to assure manufacturability.

```
thi s5 z1 > 1
thi s10 z3 > .1
thi s15 z1 > 1
```

The final constraints/targets definition is given in Fig. 6.17.

Note that the constant focus position is maintained because the back focus distance (THI s24) is not variable and the starting example has already been pre-stored with ''`PIM no`''. Whereas it is advisable to make `THI s24` variable as a non-zoomed (fixed) parameter in subsequent optimization

```
efl z1 = 5.9
efl z2 = 15
efl z3 = 59
spd f1..2 0
spd f3 0 ; wt = 0.7
oal s1..16 = 99.5
thi s5 z1 > 1
thi s10 z3 > .1
thi s15 z1 > 1
```

Figure 6.17: Final constraints/targets definition.

refinements (that would give us an additional degree of freedom to achieve better correction), we stay with the current optimization setting in order not to overload this exercise.

Now close the dialog box. We may wish to list the optimization conditions (use the command `LIS OPT` or select from the main menu *List –> Optimization –> All*).

```
Optimization setup :          File = Zoom-Lens_0.OTX
Variables :
   1   CUY
   2   CUY
   3   CUY
   4   CUY
   5   CUY
   6   CUY
   7   CUY
   8   CUY
   9   CUY
  10   CUY
  11   CUY
  12   CUY
  13   CUY
  14   CUY
  15   CUY
  17   CUY
  18   CUY
  19   CUY
  20   CUY
  21   CUY
  22   CUY
  23   CUY
  24   CUY

Zoom Variables :
   5   THI
  10   THI
  15   THI

Targets and constraints :
  efl z1 = 5.9
  efl z2 = 15
```

```
efl z3 = 59
spd f1..2 0
spd f3 0 ; wt = 0.7
oal s1..16 = 99.5
thi s5 z1 > 1
thi s10 z3 > .1
thi s15 z1 > 1
```

The optimization can now be run. In the command line, type `OPT` or select from the main menu *Optimization –> Optimize*. The merit function should decrease as the optimization evolves:

```
KT Optimization:
Number of Zoom variables         :    9
Number of variables              :   32
Number of functions              : 2430
Number of equality constraints   :    6
Number of inequality constraints :    3

Targets/Constraints                     Target      Function         Error
efl z1 = 5.9                          5.900000      5.840701     -0.059299
efl z2 = 15                          15.000000     14.760537     -0.239463
efl z3 = 59                          59.000000     57.560921     -1.439079
spd f1..2 0                           0.000000      0.016804      0.016804
spd f3 0 ; wt = 0.7                   0.000000      0.026450      0.026450
oal s1..16 = 99.5                    99.500000     98.805174      1.229824
thi s5 z1 > 1                         1.000000      1.330000      0.330000
thi s10 z3 > .1                       0.100000      0.100000      0.000000
thi s15 z1 > 1                        1.000000      0.430000     -0.570000

Iter           Min.          Equal.         Inequal.        DumpingF.      Improv.
   0       0.660490        1.974295         0.754983      1.000000
   1       0.235616        0.938567         0.574456     0.1000000E-01     0.64327
   2       0.128306        0.041125         0.000000     0.1000000E-02     0.45545
   3       0.096400        0.037822         0.000000     0.1000000E-03     0.24867
   4       0.086081        0.021050         0.000000     0.3148169E-03     0.10705
   5       0.083836        0.008540         0.000000     0.3148169E-03     0.02607
   6       0.082402        0.008073         0.000000     0.3148169E-03     0.01711
   7       0.081426        0.008142         0.000000     0.3148169E-03     0.01184
   8       0.079383        0.008972         0.000000     0.3148169E-03     0.02510
   9       0.078511        0.007531         0.053472     0.3148169E-03     0.01097
  10       0.077906        0.007452         0.000000     0.3148169E-03     0.00771

Optimization stopped. Improvement is less than    0.01000 (1.00%)

Targets/Constraints                     Target      Function         Error
efl z1 = 5.9                          5.900000      5.900001      0.000001
efl z2 = 15                          15.000000     14.999983     -0.000017
efl z3 = 59                          59.000000     58.999880     -0.000120
spd f1..2 0                           0.000000      0.001808      0.001808
spd f3 0 ; wt = 0.7                   0.000000      0.003582      0.003582
oal s1..16 = 99.5                    99.500000     99.500000      0.000000
thi s5 z1 > 1                         1.000000      1.000000      0.000000
thi s10 z3 > .1                       0.100000      0.100000      0.000000
thi s15 z1 > 1                        1.000000      1.000000      0.000000
```

The improvement in performance (that is the reduction of aberrations) is evident as seen from the transverse ray aberration plot (Fig. 6.18). We have finished this optimization example and it is recommended to save the design.

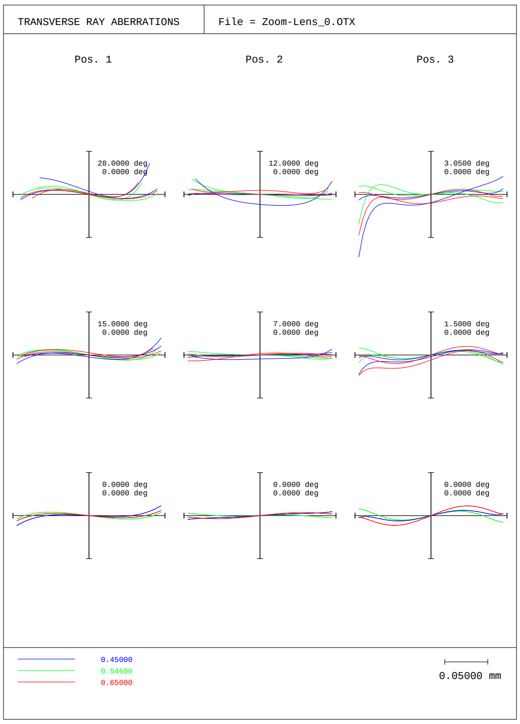


Figure 6.18: Transverse ray aberrations after 10 optimization cycles. Compare also with Fig. 6.13

7

Coatings

7.1 Basics

7.2 Formula Entry

7.3 Coating Analysis

This page is blank.

8

Ghost Analysis

This chapter shows how to analyze ghost effects in lenses due to unwanted double internal reflections and how to control it. All lens surfaces reflect light to an extent depending on the refractive index of the glass itself respectively on the type of anti-reflection coating applied to these surfaces. Light reflected from the inner surfaces of a lens will be reflected again and may form reasonably well-defined images close to the image surface. Such spurious images are called ghost images. *OpTaliX* provides several methods and techniques that help finding spurious light effects seen in the image plane. We demonstrate ghost effects on the example of an ultra wideangle lens, a Nikon fisheye lens, that is known to produce a pronounced ring shaped reflex. The design file is found in the examples library under the category "Ghost".

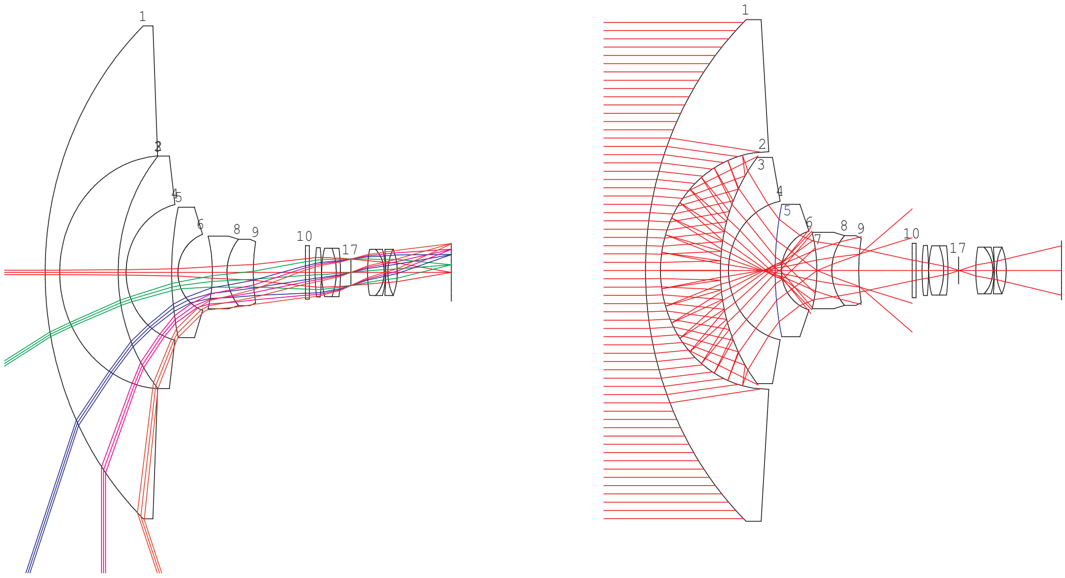


Figure 8.1: Lens layout of a 210 degree fisheye lens. Left: Ray geometry for normal imaging. Right: Ray geometry for ghost imaging, where the first ghost reflection occurs at surface 3, the second ghost reflection at surface 2.

The lens design shown in Fig. 8.1 is modelled by two zoom (multiconfiguration) positions, where the first position reflects the normal imaging conditions and position 2 is for optimum ghost imaging. The peculiarity of this design is that the entrance pupil is relatively small (approx. 2mm in diameter) whereas the front lens diameter is huge (approx. 204mm diameter) in comparison. Thus, for the sake of ghost analysis, an entrance beam of 204mm or greater must be provided.

This is adjusted in position 2.

In order to analyze ghost effects we must define the working parameters for this analysis type. We define the surfaces where the first and second internal reflection takes place by the command

```
GHO SUR s2..3
```

that is, the ghost rays are firstly reflected at surface 3, travel back to surface 2 where they are reflected a second time, and they travel forward in direction of the target surface, which is typically the image surface, if they are not blocked elsewhere. The target surface is defined by

```
GHO TAR s24
```

These parameters may also conveniently defined using a dialog, which is invoked by the command

```
GHO RGB ?
```

or from the main menu under "Geom.Analysis" - "Ghost Analysis" - "Ghost RGB Image". The dialog is shown in Fig. 8.2.

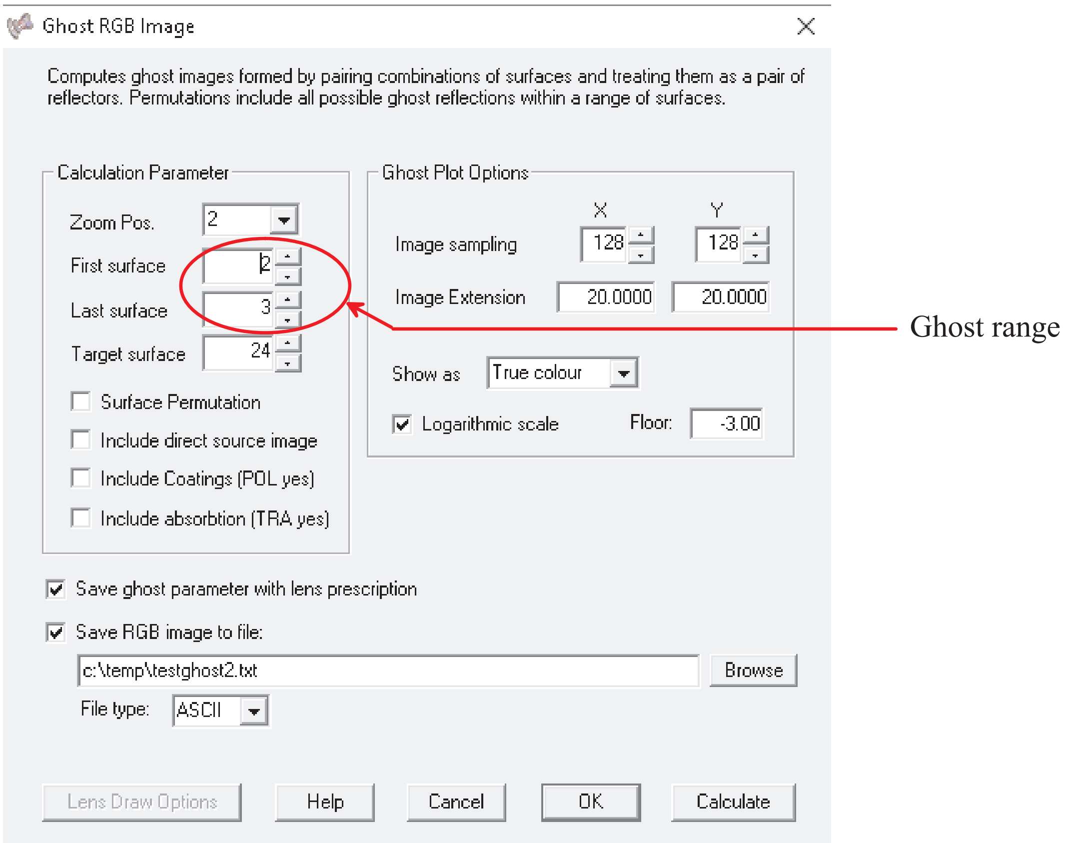


Figure 8.2: Dialog for editing the ghost working parameters.

By pressing the "calculate" button in this dialog (Fig. 8.2) a bundle of ghost rays are traced through the system for this surface combination. Fig. 8.3 shows the resulting ghost image in form of a pronounced ring-shaped reflex. We note some important observations to this analysis: Since this ring-shaped reflex only occurs in a narrow range in the aperture coordinates, the density of rays producing the effect is low. Hence, we must increase the rays across diameter in the entrance pupil

(NRD command) to minimum of NRD 128, or better yet, NRD 256. This modification increases the probability that enough rays will be traced that will produce the desired reflex. At this point the reader may raise the objection that many more possible surface pairs may produce ghosts, other than the single surface pair 2-3, which we have analyzed so far. This is true and in fact, the number of possible surface combinations (pairs) which may contribute to ghost images is $n(n-1)/2$, where n is the number of lens surfaces in the system. We can account for this condition if we check the box "Surface permutation" and set the ghost surface range to the first and last surface in the system in Fig. 8.2. Then the program calculates all possible surface pairs successively.

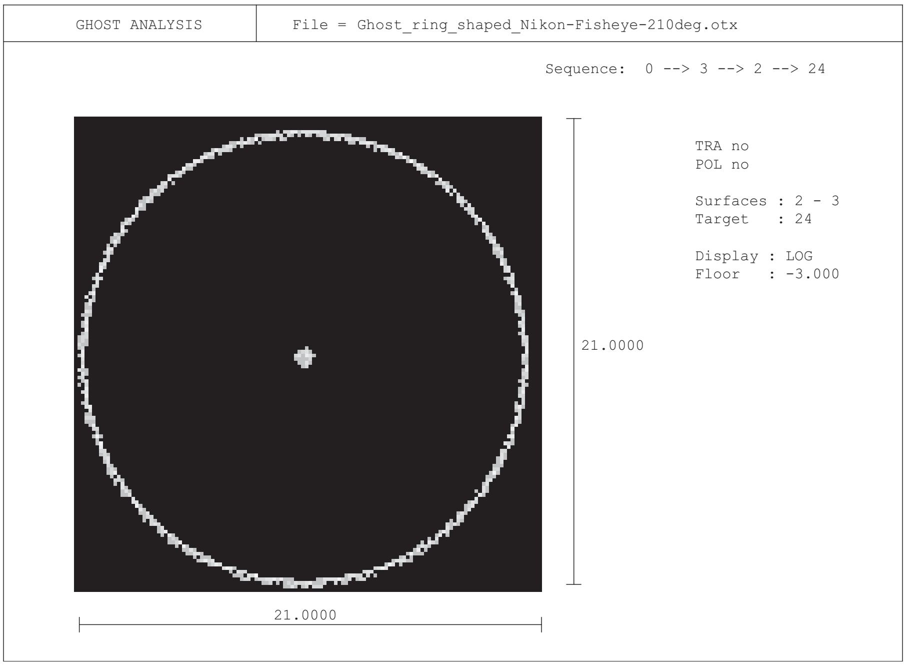


Figure 8.3: Ghost image showing a pronounced ring-shaped reflex. The analysis was done using a 512x512 ray grid in the entrance pupil.

8.1 Control of Ghost Effects

In this section we want to control the ghost reflex created by the surface pair 2-3, which we have seen in the previous section. Controlling the ghost reflex means minimizing or even eliminating the disturbing effect. We have seen that this ring shaped ghost is produced by rays impinging into the system in a very narrow aperture region. In order to visualize the suspected ghost rays we use the "plot a single ray" option in the main menu under "Display", or available by the command RSP. This option invokes a dialog with slider bars and, in conjunction with the ghost lens layout option (use command GHV ?), we can easily identify the coordinates of the offending ray. Fig. 8.4 shows the ghost ray (doing the double reflection at surfaces 2-3), the slider bar dialog indicates that the relative start coordinates in the entrance pupil are at 0.3 * EPD, where EPD is the pupil diameter at zoom position 2, thus the ray starts at Y = 0.3*204/2 = 30.6mm absolute.

Now we can perform a numerical raytrace to get the image coordinate of the ghost ray. The command

Figure 8.4: Slider bar dialog to visualize the offending ghost ray.

`GHRSI`

is identical to the `RSI` command (`RSI` = Relative Single ray), which traces a single ray without ghost reflections. The `GHRSI` variant, however, traces a single ray including the ghost reflections. Specifically in this example design, we issue the command

`ghrsi f1 z2 w1 0 0.3`

and get the result:

```
Ghost Ray intersection coordinates :
**********************************

Zoom position =   2

Field  =  1      OBX =      0.00000   OBY =      0.00000
Colour =  1      WL =       587.000 nm
        #         X          Y          Z         CX          CY          CZ      cos(I)
REL     1   0.00000   30.60000    3.42304  0.0000000  -0.0765511  0.9970657  0.975282
REL     2   0.00000   29.57487   10.10278  0.0000000   0.3213436  0.9469627  0.835520
REL     3   0.00000   39.06243   11.37170  0.0000000   0.3213436  0.9469627  0.626516
GREL    2   0.00000   47.03354   37.17711  0.0000000   0.3213436  0.9469627  0.835520
REL     3   0.00000   41.07556   12.69970  0.0000000  -0.8361162  0.5485524  0.816470
REL     4   0.00000   26.85777   18.40541  0.0000000  -0.7692125  0.6389931  0.977736
REL     5   0.00000   21.68415    1.92324  0.0000000  -0.5776850  0.8162598  0.764401
REL     6   0.00000   14.71167    8.91563  0.0000000  -0.3195043  0.9475848  0.889848
REL     7   0.00000   12.90398   -1.45107  0.0000000  -0.1039055  0.9945872  0.852959
REL     8   0.00000   11.67605    3.63024  0.0000000  -0.1374108  0.9905142  0.878165
REL     9   0.00000   10.41444    0.71404  0.0000000  -0.1380863  0.9904202  1.000000
REL    10   0.00000    7.19153    0.00000  0.0000000  -0.0909644  0.9958541  0.990420
REL    11   0.00000    7.03481    0.00000  0.0000000  -0.1380863  0.9904202  0.995854
REL    12   0.00000    6.62655    0.25927  0.0000000  -0.1145450  0.9934181  0.998182
REL    13   0.00000    6.35029   -0.20444  0.0000000  -0.2316000  0.9728111  0.983993
REL    14   0.00000    6.15802    0.50784  0.0000000  -0.2080945  0.9781087  0.997610
REL    15   0.00000    4.94508   -0.46340  0.0000000  -0.1532041  0.9881946  0.922419
REL    16   0.00000    4.59915   -0.13853  0.0000000  -0.3201629  0.9473625  0.977182
REL    17   0.00000    2.94164    0.00000  0.0000000  -0.3201629  0.9473625  0.947363
REL    18   0.00000    0.36407    0.00135  0.0000000  -0.2135338  0.9769357  0.949719
REL    19   0.00000   -1.28890   -0.06183  0.0000000  -0.1953849  0.9807266  0.992889
REL    20   0.00000   -1.40924   -0.02971  0.0000000  -0.3166310  0.9485488  0.988090
REL    21   0.00000   -1.44780   -0.00952  0.0000000  -0.1900473  0.9817749  0.952630
REL    22   0.00000   -1.64733    0.06805  0.0000000  -0.2264642  0.9740195  0.962756
```

```
REL  23    0.00000    -2.60761    -0.13804   0.0000000  -0.2883320   0.9575305   0.992486
REL  24    0.00000   -10.07127     0.00000   0.0000000  -0.2883320   0.9575305   0.957530
```

The X,Y,Z coordinates of the ghost rays are accessible via LDI (lens data item), so the Y-coordinate is evaluated as

`eva [ghy s24 z2 f1 w1 0 0.3]`

The program will return the value -10.07127. We are now prepared to use this result in the optimization constraint setting (merit function). The optimization constraints in this example are related to zoom position 1 only (the imaging part) and are kept simple. Without further explanation we note the constraints `efl, spd` and `y` to control the focal length (`efl`), the imaging properties (spot diameter `spd`) and the field heights (`y`) in position 1. We will add a constraint, in which we will attempt to move the Y-coordinate of the ghost ray (GHY) outside of the active image surface. The constraint is

`ghy s24 z2 f1 w1 0 0.3 < -24`

and is marked yellow in the variables/targets dialog in Fig. 8.5.

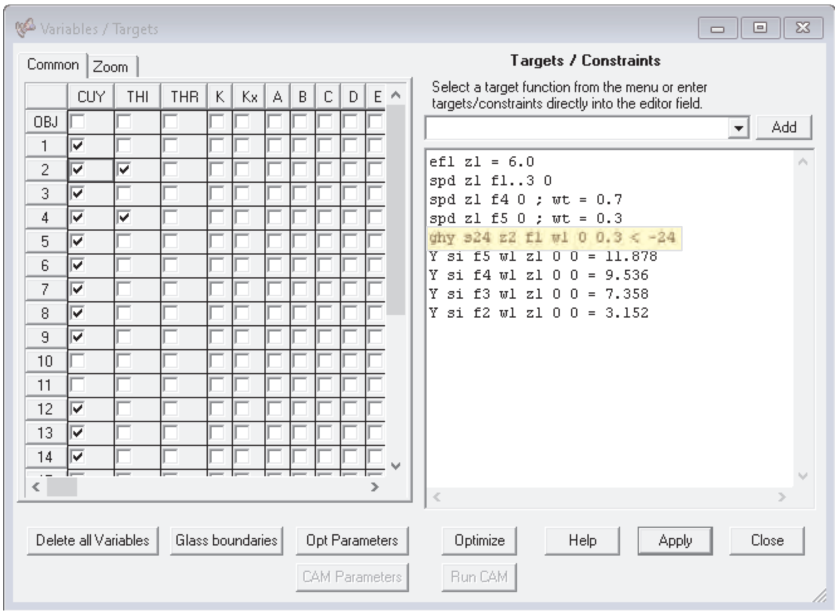


Figure 8.5: Optimization constraints with the addition of ghost ray Y-coordinate (GHY) requirement, marked in yellow.

In the optimization constraints dialog we do an optimization by pressing the "Optimize" button three times in succession (doing the optimization thrice), and subsequently redo the ghost RGB analysis. Fig. 8.6 shows the resulting ghost image, which should be compared with Fig. 8.3.

Figure 8.6: Optimization constraints with the addition of ghost ray Y-coordinate (GHY) requirement, marked in yellow.

Bibliography

[1] Aberrator: http://www.aberrator.astronomy.net/

[2] J.A. Arnaud, ''Hamiltonian Theory of Beam Mode Propagation'', Progress in Optics, Vol.11, E. Wolf Ed. (Amsterdam 1973)

[3] M.Berek, ''Grundlagen der praktischen Optik'', Walter de Gruyter & Co. , Berlin 1970, Nachdruck der 1.Auflage 1930

[4] M.Born, E.Wolf, ''Principles of Optics'', 6^{th} edition, Pergamon Press

[5] Breault Research Organization, http://www.breault.com/

[6] E.Oran Brigham. ''FFT, Schnelle Fouriertransformation'', Oldenburg Verlag, München,Wien 1982

[7] H. A. Buchdahl, ''Many-color correction of thin doublets'', Applied Optics, Vol. 24, No. 12, June 1985

[8] Cargille Laboratories Inc., 55 Commerce Rd., Cedar Groove, N.J. 07009, USA

[9] A.E. Conrady, Mon. Not. Royal astr. Soc., 64 (1904) 182

[10] J.E. Dennis Jr., R.B. Schnabel, ''Numerical Methods for Unconstrained Optimization and Nonlinear Equations'', Prentice-Hall, Englewood Cliffs, New Jersey (1983)

[11] Donald P. Feder, ''Automatic optical design'', Appl. Opt., Vol.2, No.12, Dec. 1963, pp. 1209

[12] W.G.Driscoll, W.Vaughan, ''Handbook of Optics'' Mc.Graw Hill, 1978, `ISBN 0-07-047710-8`

[13] A. Ghatak, K. Thyagarajan, ''Optical Electronics'', Cambridge University Press, 1989, ISBN 0 512 30643 4

[14] P.E. Gill, W.Murray, M.A. Sanders and M.H. Wright, ''Model building and practical aspects of nonlinear programming'', Computational Mathematical Programming (edited by K. Schittkowski), NATO ASI series, 15, Springer-Verlag, Berlin, Germany (1985)

[15] D. Gloge, ''Weakly Guiding Fibers'', Appl.Opt., Vol.10, No.10, Oct.1971, pp.2252-2258

[16] D. S. Grey, ''The inclusion of tolerance sensitivities in the merit function for lens optimization'', SPIE Vol. 147, Computer-Aided Optical Design (1978).

[17] J.W. Goodman, ''Introduction to Fourier Optics'', McGraw-Hill, New York, (1988), $2^{nd} edition$

[18] H.Haferkorn. ''Optik, Physikalisch technische Grundlagen und Anwendungen''. Harri Deutsch, Thun, Frankfurt/M. 1981

[19] H.Haferkorn, ''Bewertung optischer Systeme'', VEB Deutscher Verlag der Wissenschaften, Berlin 1986, ISBN 3-326-00000-6

[20] R.Herloski, S.Marshall, R.Antos, ''Gaussian beam ray-equivalent modeling and optical design'', Appl.Opt. 22, No.8, pp. 1168-1174, April 1983

[21] J.L.Houghton, US Patent 2,350,112 (30.May 1944)

[22] R.A. Hull, ''Transmission of light through a pile of parallel plates'', Proc.Phys.Soc., 574-575 (1936)

[23] M.Isshiki et.al., ''Lens Design: Global optimization of both performance and tolerance sensitivity'', SPIE, Vol. 6342, International Optical Design Conference 2006

[24] M.J. Kidger, ''Intermediate Optical Design'', SPIE Press monograph, PM 134, ISBN 0-8194-5217-3

[25] T.Kita et.al., Appl.Opt., Vol. 22, No.4, pp. 512

[26] W.A. Kleinhans, ''Aberrations of curved zone plates and Fresnel lenses'' Appl. Opt. 16, pp. 1701-1704, (1977)

[27] H.W. Kogelnik, T.Li, Appl.Opt. 5, 1550 (1966)

[28] F. Kohlrauch, ''Praktische Physik'', 1968, Vol.1, p.408.

[29] M. Laikin, ''Lens Design'', 3^{rd} edition, Marcel Dekker Inc., (2001)

[30] K. Levenberg, ''A method for the solution of certain problems in least squares'', Quarterly of Applied Mathematics, **2**, 164-168

[31] R.J.Lurie, ''Anastigmatic Catadioptric Telescopes'', Journal of the optical society of America (JOSA), Vol.65(3), 1975, p261-266,

[32] H.A. Macleod, ''Thin-film optical filters'', 2^{nd}Ed., Adam Hilger Ltd., Bristol ISBN 0-85274-784-5

[33] D.Malacara, ''Diffraction Performance Calculations in Lens Design'', SPIE, Vol.1354, June 1990, pp.2-14

[34] D.Malacara, ''Handbook of Lens Design'', Marcel Deccer, Inc., (1994), ISBN 0-8247-9225-4

[35] D.Marquardt, ''An algorithm for least-squares estimation of nonlinear parameters'' SIAM Journal on Applied Mathematics, **11**, 431-441

[36] M.Mansuripur, ''Classical Optics and its Applications'', Cambridge University Press, (2002), ISBN 0 521 80093 5

[37] R. I. Mercado, ''Design of apochromats and superachromats'', SPIE Critical Review Vol. CR41, Lens Design, ed. Warren J.Smith (Jan 1992).

[38] R.C.Millard, G.Seaver, ''An index of refraction algorithm for seawater over temperature, pressure, salinity, density and wavelength'', Deep-Sea Research, Vol. 37, No.12, pp.1909-1926, (1990)

[39] NSG Website, www.nsgeurope.com/technical.shtml

[40] OpsiraGmbH, Leibnitzstr. 20, D-88250 Weingarten, Germany, http://www.opsira.de

[41] Matthew P. Rimmer, ''Ray tracing in inhomogeneous media'', SPIE Proc. 399, p.339, (1983)

[42] Matthew P. Rimmer, ''Relative illumination calculations'', SPIE Proc. 655, p.99, (1986)

[43] M.J.D. Powell, ''A fast algorithm for nonlinearly constrained optimization calculations'', Numerical Analysis Proceedings, Dundee 1977, Lecture Notes in Mathematics (edited by G.A. Watson), 630, pp.144-157, Springer-Verlag, Berlin, Germany

[44] Radiant Vision Systems, http://www.radiantvisionsystems.com

[45] J.L. Rayces, L.Lebich, ''Modeling of diffractive optical elements for optical design'', SPIE Vol. 2000, pp. 43, 0-8194-1249-X/93

[46] P.N.Robb, R.I.Mercado, ''Calculation of refractive indices using Buchdahl's chromatic coordinate'', Appl.Optics, Vol.22, No.8, April 1983, pp.1198

[47] J.B.Scarborough, ''Numerical Mathematical Analysis'', Johns Hopkins U.P., Baltimore, 1966; Chap.13, Article 116.

[48] SCHOTT Glaswerke, ''Dispersionsformel fuer die Temperaturkoeffizienten der Brechzahlen von Glaesern''. Technische Information Nr. 19 (4/1988) der Schott Glaswerke, Postfach 2480, D-6500 Mainz.

[49] A.Sharma, D.V.Kumar, A.K.Ghatak, ''Tracing rays through graded-index media: a new method'', Applied Optics, Vol.21, No. 6, March 1982

[50] A.E. Siegman, Proc. SPIE 1224, 2, 1990

[51] Gordon H. Spencer, ''A flexible automatic lens correction procedure'', Appl. Opt., Vol.2, No.12, Dec. 1963, pp. 1257

[52] W.C. Sweatt, ''Describing holographic optical elements as lenses'' J.Opt.Soc.Am. 67, pp.803-808, 1977

[53] W.C. Sweatt, ''Mathematical equivalence between a holographic optical element and an ultra-high index lens'' J.Opt.Soc.Am. 69, No.3, pp.486-487, 1979

[54] Y.Tamagawa, S.Wakabayashi, T.Tajime, ''New design method for athermalized optical systems'', SPIE, Vol. 1752 (1992)

[55] Y. Tamagawa et.al., ''Multilens system design with an athermal chart'', Appl.Opt., Vol. 33, No. 34, pp. 8009, Dec. 1994

[56] Y. Tamagawa, T. Tajime, ''Expansion of an athermal chart into a multilens system with thick lenses spaced apart'', Opt. Eng. 35(10), pp. 3001, Oct. 1996

[57] R.E. Wagner, W. J. Tomlinson, ''Coupling efficiency of optics in single-mode fiber components'', Appl.Optics, Vol.21, pp. 2671-2688 (1982)

[58] W.T.Welford, ''A vector raytracing equation for hologram lenses of arbitrary shape''. Optics Communications, Vol.14, No. 3, July 1975, p.322

[59] E.Waluschka, ''Polarization Ray Trace'', Optical Engineering, Feb. 1989, Vol.28, No.2

[60] J.Y.Wang, D.E.Silva, ''Wave-front interpretation with Zernike polynomials'', Applied Optics, Vol. 19(9), pp. 1510-1518, May 1980

[61] R.N. Wilson, ''Reflecting Telescope Optics I'', Springer Verlag Berlin-Heidelberg 1996, ISBN 3-540-58964-3

[62] T.R.M.Sales, G.M.Morris, ''Diffractive-refractive behaviour of kinoform lenses'', Appl.Opt., Vol.36, No.1, January 1997, pp. 253

[63] G.W.Forbes, ''Shape specification for axially symmetric optical surfaces'', OSA, Vol.15, No.8, April 2007, Optics Express 5218

[64] G.W.Forbes, ''Robust and fast computation for the polynomials of optics'', OSA, Vol.18, No.12, June 2010, Optics Express 13851

[65] C.K. Carniglia, ''Accounting for the phase of an optical coating in an optical system'', OSA, Opticsinfobase

[66] MacLeod: Optical Thin Films, www.thinfilmcenter.com

[67] Masaki Isshiki, "Global optimization with escape function". Proc. SPIE 3482, International Optical Design Conference 1998, Kona, HI, USA.

Index

Symbols

A

B

C

D

E

F

G

H

K

L

M

N

O

P

Q

R

S

Y

Z