OpTaliX®

Software for Optical Design and Thin Films

Tutorial

Version 6.50

Optenso™
Optical Engineering Software

support@optenso.de
http://www.optenso.de/

June 11, 2006

Copyright

Copyright 1986 - 2006 by Optϵnso®. All rights reserved worldwide. No part of this manual and the software may be copied, distributed, transcribed, transmitted, or stored in a retrieval system by any means without express written permission from the author. The information in this manual and the software it describes are subject to change without notice.

Trademarks

OpTaliX® is a registered trademark of Optϵnso®.

Microsoft Windows, Windows 95, Windows 98, Windows NT, Windows 2000 and Windows XP are trademarks of Microsoft Corporation.

OSLO is a registered trademark of Sinclair Optics, Inc.

ZEMAX is a registered trademark of Zemax Development Corp.

CODE V is a registered trademark of Optical Research Associates

POV-Ray^{TM} is Copyright by the POV-Ray Team^{TM}

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.

Optεnso™
Optical Engineering Software

Herbstweg 9
86859 Igling
Germany
E-mail : info@optenso.de
Web : http://www.optenso.de/

Contents

1

Installation

OpTaliX is installed by a single install program. The install program recognizes previous installations of *OpTaliX* on your computer and will backup (on user request) older files. Installation is started by :

- Insert the *OpTaliX* CD-ROM in the CD-ROM drive.
- Open the Explorer and select the CD-ROM drive
- Double click the file `optalix_pro_setup.exe`
- The installation starts, follow the instructions on the screen.

1.1 Software Requirements

A 32-bit Microsoft Windows operating system is required to run *OpTaliX* . Supported versions are Windows 95, Windows 98, Windows NT, Windows 2000 and Windows XP. We explicitly do NOT support Windows-Me.

1.2 Hardware Requirements

minimum:	recommended:
Pentium 100MHz	Pentium 4, 1GHz or faster
64Mbyte RAM	256MB
60Mbyte available on hard disc	80Mbyte available on hard disc
800 x 600 pixel monitor resolution	1024 x 768 pixel monitor resolution or higher

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 98/Me/NT/2000/XP, click the **Start** button, click **Programs**, click the *OpTaliX* program group, and then click the *OpTaliX* menu item, as shown in Figure 2.1.

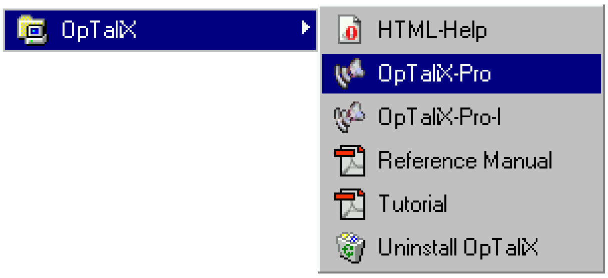


Figure 2.1: *OpTaliX* program group menu.

The *OpTaliX* program group also includes menu items for HTML-Help, Reference Manual, Tutorial and uninstalling *OpTaliX* . Note that two menu items for *OpTaliX* are found: **OpTaliX-Pro** and **OpTaliX-Pro-I**. Both versions, OpTaliX-Pro and OpTaliX-Pro-I, are functionally identical, except for the style of the windows.

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 start *OpTaliX* by typing

```
C:> c:\programs\optalix\optalixp mylens.otx
```

or

```
C:> c:\programs\optalix\optalixp-i mylens.otx
```

depending on the version (floating windows or inside root windows) to be used. If *OpTaliX* was installed in a different directory than `c:\programs\optalix`, the path to the *OpTaliX* executable must be accordingly modified. 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 loaded 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.
- Select the main window (click on the title bar of the main window) and press the ESC-key.
- In the command line, type `EXI` or `QUIT` and press Return.

In all cases, you will be asked to confirm the exit. After you exit *OpTaliX* , you are 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

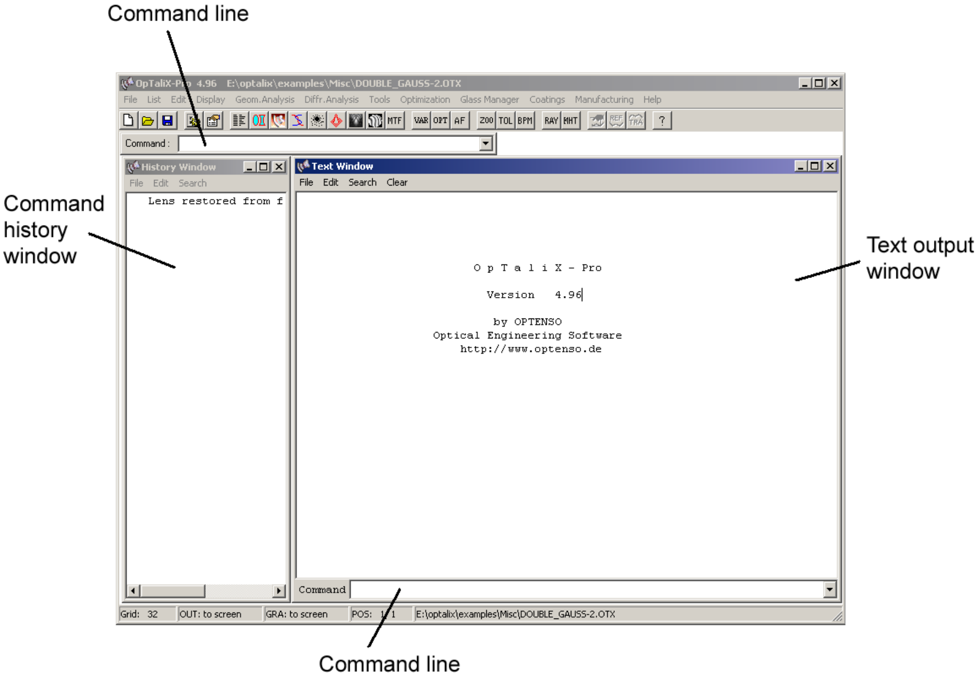
```
EXI Yes
```

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.


OpTaliX-Pro 4.96 E:\optalix\examples\Misc\DOUBLE_GAUSS-2.OTX
File List Edit Display Geom.Analysis Diffr.Analysis Tools Optimization Glass Manager Coatings Manufacturing Help
MTF VAR OPT AF ZOO TOL BPM RAY MHT ?
Command :
History Window
File Edit Search
Text Window
File Edit Search Clear
Command
Grid: 32
OUT: to screen
GRA: to screen
POS: 1 1
E:\optalix\examples\Misc\DOUBLE_GAUSS-2.OTX

```
Lens restored from f
```

```
O p T a l i X - Pro

Version   4.96

by OPTENSO
Optical Engineering Software
http://www.optenso.de
```

Command line

Command history window

Text output window

Command line

Figure 3.1: $OpTaliX$ main window. Note that size and position of the windows may be different from this figure.

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 underneath the toolbar and in the text output window, respectively. See the reference manual for a full description of all commands available.

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.

The functions of the text window and the history window as shown in Fig. 3.1 are as follows:

Text Window The text window reports all numerical analysis results 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 automatically generated by the program.

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, tool bar and the command line.

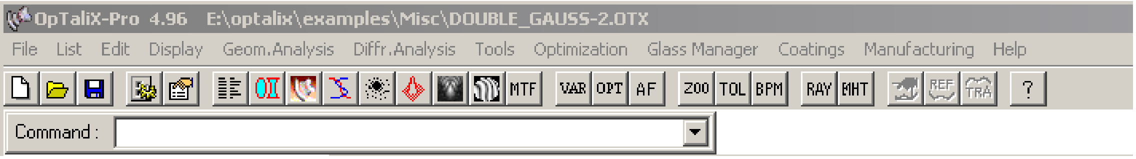


Figure 3.2: Main window, containing menu bar, tool bar and upper command line.

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 asphérization 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

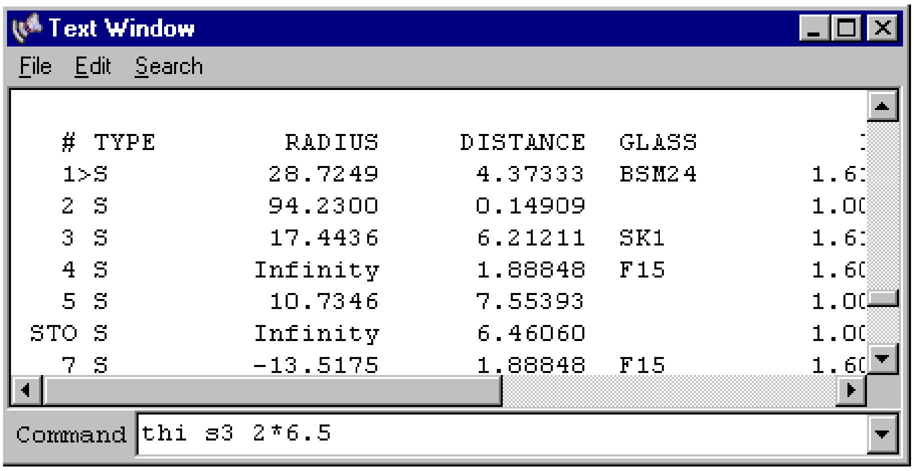
The tool bar provides quick access to perform often used tasks. A concise online description is shown for each icon if the mouse cursor is held over the icon.

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

	Restores a system from a file.
	Saves system data to a 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 the spot diagram.
	Calculates diffraction point spread function (PSF).
	Calculates a grey-shaded picture of the PSF.
MTF	Calculates diffraction MTF vs. field. If only one field point is specified, MTF vs. frequency is computed instead.
VAR	Edit optimization variables and constraints.
OPT	Execute the optimization.
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.)
	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

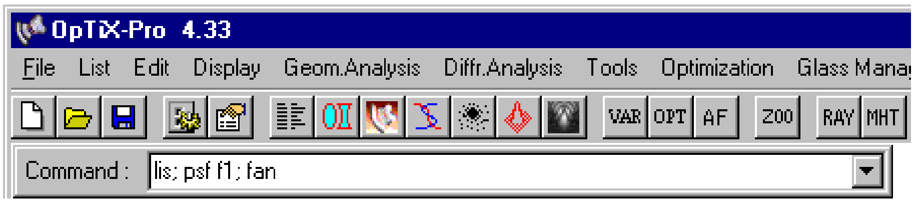
In addition to using menus, *OpTaliX* can be run using only commands. Two command lines are provided, one underneath the toolbar and one at the bottom of the text window, as shown in Figs. 3.3 and 3.4. They are fully equivalent in functionality. The commands and their detailed syntax are described in 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.



```
  # TYPE          RADIUS     DISTANCE   GLASS
  1>S            28.7249      4.37333   BSM24        1.6
  2 S            94.2300      0.14909                1.00
  3 S            17.4436      6.21211   SK1          1.6
  4 S           Infinity      1.88848   F15          1.60
  5 S            10.7346      7.55393                1.00
STO S           Infinity      6.46060                1.00
  7 S           -13.5175      1.88848   F15          1.60
```

Command thi s3 2*6.5

Figure 3.3: Command line at the bottom of the text output window. Note the usage of numeric expression for parameters.


OpTiX-Pro 4.33
File List Edit Display Geom.Analysis Diffr.Analysis Tools Optimization Glass Mana
VAR OPT AF ZOO RAY MHT

Command : lis; psf f1; fan

Figure 3.4: Command line underneath the main menu and toolbar. Note the use of several commands on a single line, separated by semicolons.

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

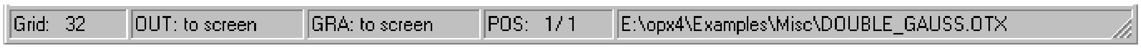
For example, `s3..4` denotes surfaces 3 to 4. If a command is entered with no arguments, default values will be used where possible. Some commands use settings in the configuration dialog. For example, the spot diagram uses the values for field and wavelength as defined in the configuration.

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
```

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: 32 | OUT: to screen | GRA: to screen | POS: 1/1 | E:\opx4\Examples\Misc\DOUBLE_GAUSS.OTX

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.

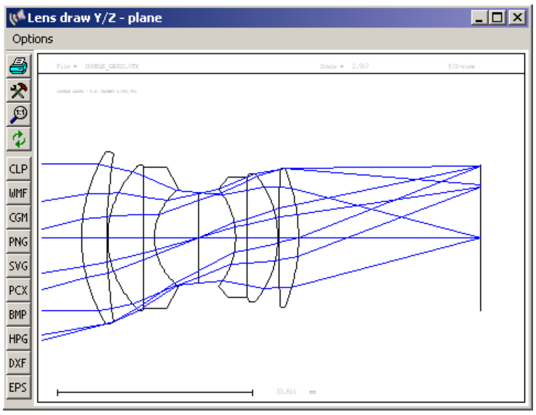


Figure 3.5: Graphics window

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 E**x**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. 3.6). releasing the mouse button will instantly display the enlarged plot.

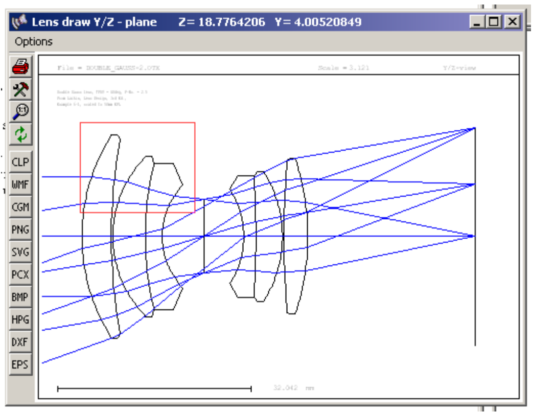


Figure 3.6: Enlarging a plot area by dragging a rectangle with the mouse.

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.

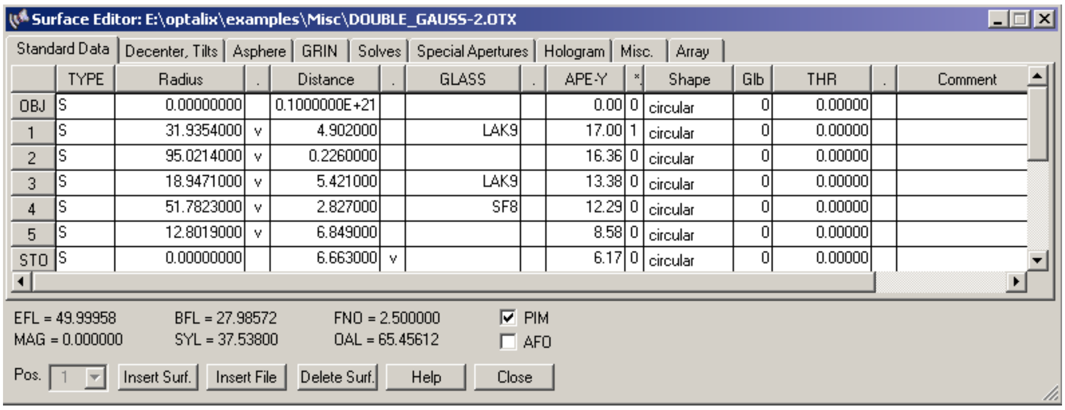
3.3 Clipboard Support

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 CLP 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. 3.7):



	TYPE	Radius	.	Distance	.	GLASS	.	APE-Y	*	Shape	Glb	THR	.	Comment
OBJ	S	0.00000000		0.1000000E+21				0.00	0	circular	0	0.00000		
1	S	31.9354000	v	4.902000		LAK9		17.00	1	circular	0	0.00000		
2	S	95.0214000	v	0.2260000				16.36	0	circular	0	0.00000		
3	S	18.9471000	v	5.421000		LAK9		13.38	0	circular	0	0.00000		
4	S	51.7823000	v	2.827000		SF8		12.29	0	circular	0	0.00000		
5	S	12.8019000	v	6.849000				8.58	0	circular	0	0.00000		
STO	S	0.00000000		6.663000	v			6.17	0	circular	0	0.00000		

EFL = 49.99958 BFL = 27.98572 FNO = 2.500000 ☑ PIM
MAG = 0.000000 SYL = 37.53800 OAL = 65.45612 □ AFO

Figure 3.7: 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.

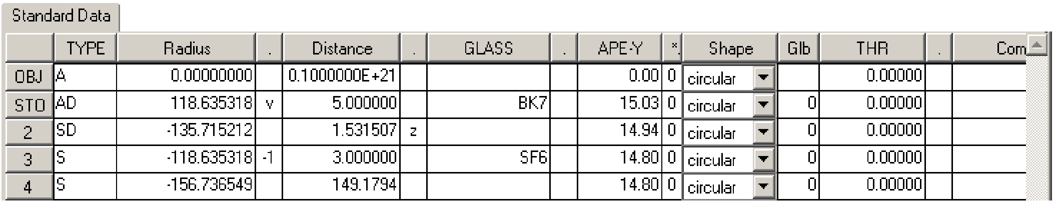
Navigating around in a spreadsheet can be accomplished with the mouse or keyboard. Mouse navigation is quite simple:

- 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 !**

Note that the surface spreadsheet editor dialog accounts for most of the resource usage of the program. This may lead to problems under Windows 95/98/Me operating systems if several other applications are simultaneously running. This is imposed by system constraints (not $OpTaliX$), since these operating systems only support a total of 16384 window handles, shared between all applications. If you encounter problems with the surface spreadsheet editor, close other applications to free resources. These limitations do not apply to Windows NT/2000/XP, which appear to be limited only by available memory.

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).

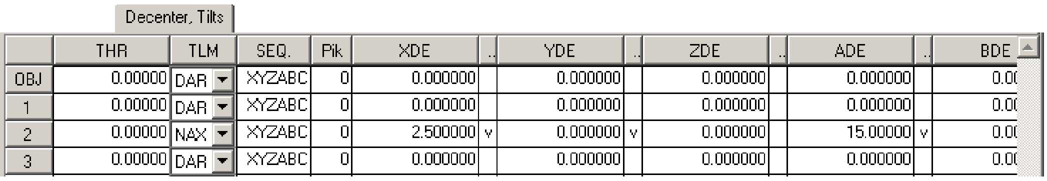
3.4.2 Decenter, Tilts

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)

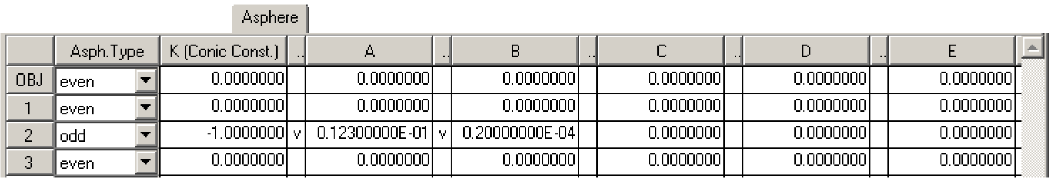
BDE Tilt about Y-axis in degree (β - tilt)

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	K (Conic Const.)	..	A	..	B	..	C	..	D	..	E
OBJ	even	0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000
1	even	0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000
2	odd	-1.0000000	v	0.12300000E-01	v	0.20000000E-04		0.0000000		0.0000000		0.0000000
3	even	0.0000000		0.0000000		0.0000000		0.0000000		0.0000000		0.0000000

Asph.Type Selects the definition of higher order coefficients. See the detailed description of ”ODD”, ”ODD30”, ”EVEN” and ”XYP” polynomial coefficients in the reference manual.

K The conic constant K describes surfaces of conic sections.

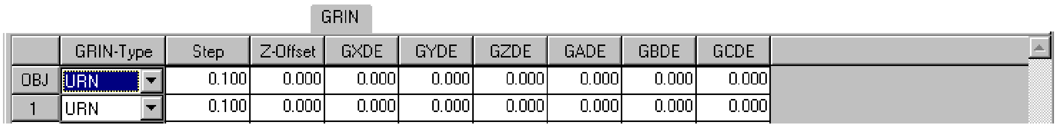
		K	$<$	-1	Hyperbola
		K	=	-1	Parabola
-1	$<$	K	$<$	0	Ellipse at major axis (prolate ellipse)
		K	$>$	0	Ellipse at minor axis (oblate ellipse)
		K	=	0	Sphere

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`).

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

	GRIN-Type	Step	Z-Offset	GXDE	GYDE	GZDE	GADE	GBDE	GCDE
OBJ	URN	0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000
1	URN	0.100	0.000	0.000	0.000	0.000	0.000	0.000	0.000

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.

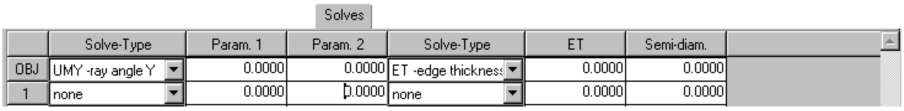
GCDE Tilt of the gradient index profile about the local surface Z-axis.

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.



	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.

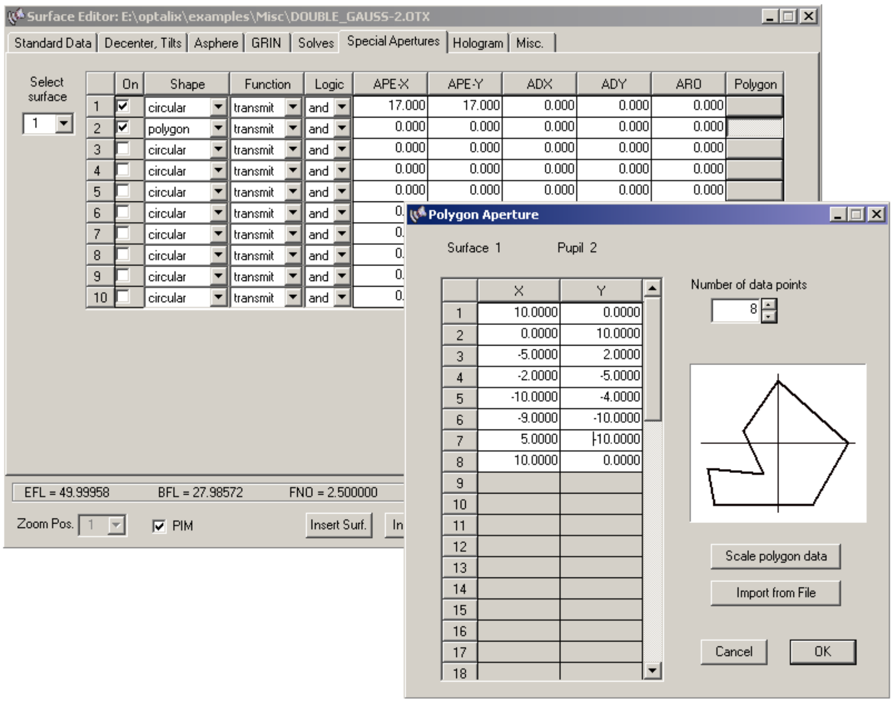
Param.1 The first parameter. It depends on the solve type. The column heading will be updated corresponding to the selected solve type to aid the user in entering the correct value.

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.



Select surface: 1

	On	Shape	Function	Logic	APE-X	APE-Y	ADX	ADY	ARO	Polygon
1	☑	circular	transmit	and	17.000	17.000	0.000	0.000	0.000	
2	☑	polygon	transmit	and	0.000	0.000	0.000	0.000	0.000	
3	☐	circular	transmit	and	0.000	0.000	0.000	0.000	0.000	
4	☐	circular	transmit	and	0.000	0.000	0.000	0.000	0.000	
5	☐	circular	transmit	and	0.000	0.000	0.000	0.000	0.000	
6	☐	circular	transmit	and	0.					
7	☐	circular	transmit	and	0.					
8	☐	circular	transmit	and	0.					
9	☐	circular	transmit	and	0.					
10	☐	circular	transmit	and	0.					

EFL = 49.99958 BFL = 27.98572 FNO = 2.500000

Zoom Pos. 1 ☑ PIM

Surface 1 Pupil 2

	X	Y
1	10.0000	0.0000
2	0.0000	10.0000
3	-5.0000	2.0000
4	-2.0000	-5.0000
5	-10.0000	-4.0000
6	-9.0000	-10.0000
7	5.0000	-10.0000
8	10.0000	0.0000
9		
10		
11		
12		
13		
14		
15		
16		
17		
18		

Number of data points: 8

Figure 3.8: Editing special apertures. A separate dialog is invoked for defining vertices of polygon apertures.

In the special apertures dialog (see Fig. 3.8) 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.

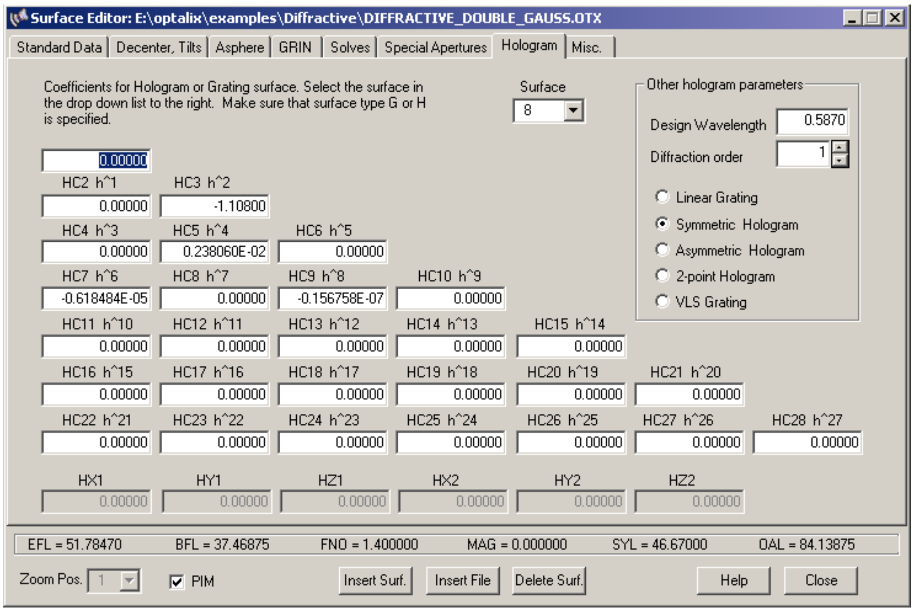
Basic apertures must be activated to be included in ray trace. Check the appropriate box in the ”On” column. If there is more than one basic aperture on a surface, the aperture data must be entered consecutively into the table.

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 40).

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.



Surface Editor: E:\optalix\examples\Diffractive\DIFFRACTIVE_DOUBLE_GAUSS.OTX

Coefficients for Hologram or Grating surface. Select the surface in the drop down list to the right. Make sure that surface type G or H is specified.

Surface: 8

0.00000

HC2 h^1: 0.00000 | HC3 h^2: -1.10800

HC4 h^3: 0.00000 | HC5 h^4: 0.238060E-02 | HC6 h^5: 0.00000

HC7 h^6: -0.618484E-05 | HC8 h^7: 0.00000 | HC9 h^8: -0.156758E-07 | HC10 h^9: 0.00000

HC11 h^10: 0.00000 | HC12 h^11: 0.00000 | HC13 h^12: 0.00000 | HC14 h^13: 0.00000 | HC15 h^14: 0.00000

HC16 h^15: 0.00000 | HC17 h^16: 0.00000 | HC18 h^17: 0.00000 | HC19 h^18: 0.00000 | HC20 h^19: 0.00000 | HC21 h^20: 0.00000

HC22 h^21: 0.00000 | HC23 h^22: 0.00000 | HC24 h^23: 0.00000 | HC25 h^24: 0.00000 | HC26 h^25: 0.00000 | HC27 h^26: 0.00000 | HC28 h^27: 0.00000

HX1: 0.00000 | HY1: 0.00000 | HZ1: 0.00000 | HX2: 0.00000 | HY2: 0.00000 | HZ2: 0.00000

Other hologram parameters

Design Wavelength: 0.5870

Diffraction order: 1

○ Linear Grating
● Symmetric Hologram
○ Asymmetric Hologram
○ 2-point Hologram
○ VLS Grating

EFL = 51.78470 BFL = 37.46875 FNO = 1.400000 MAG = 0.000000 SYL = 46.67000 OAL = 84.13875

Zoom Pos. 1 ☑ PIM

Figure 3.9: Editing hologram coefficients.

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

- 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.

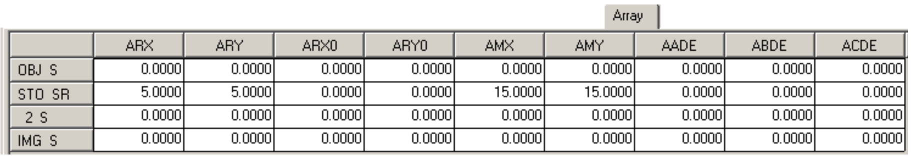
The hologram type can be selected in the right part of the dialog. Each hologram surface requires specification of the design wavelength and diffraction order.

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 grid over the channel surface. The channel centers are located at (local) X/Y coordinates defined by the X-spacing (`ARX`) and Y-spacing (`ARY`).



	ARX	ARY	ARX0	ARY0	AMX	AMY	AADE	ABDE	ACDE
OBJ S	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
STO SR	5.0000	5.0000	0.0000	0.0000	15.0000	15.0000	0.0000	0.0000	0.0000
2 S	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
IMG S	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 3.10: 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 ± 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. 3.11.

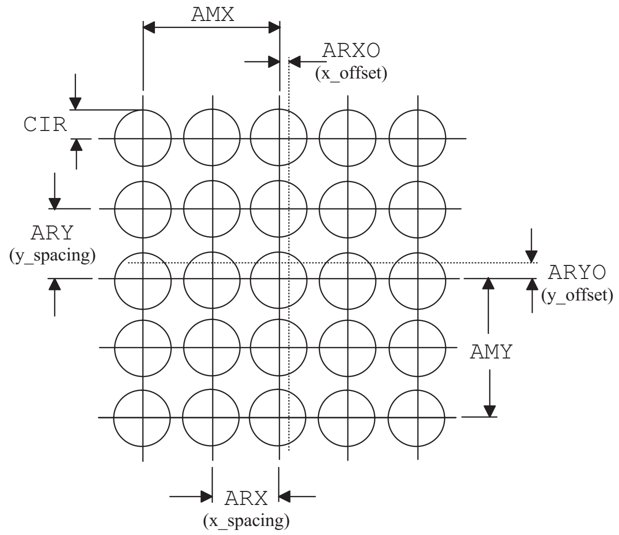


Figure 3.11: Definition of array cells (channels).

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.

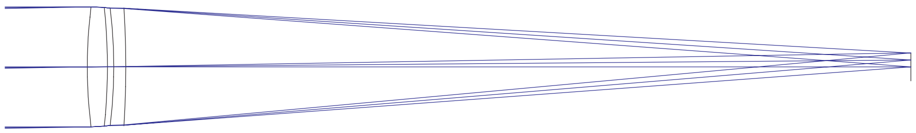


Figure 3.12: Example achromatic doublet.

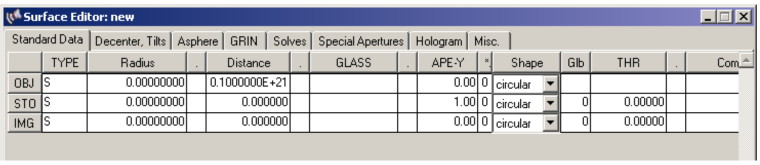
The course examples will be given in two forms, first by using menus and dialogs and then using commands only.

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

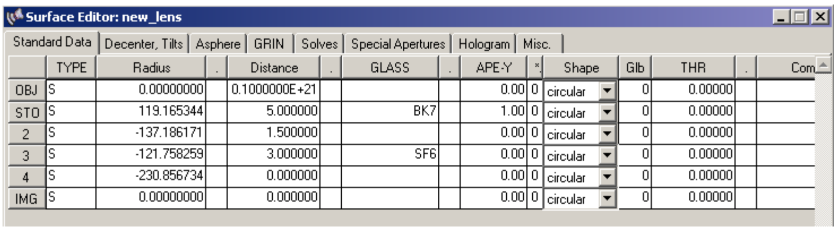
. The table contains three rows, corresponding to the object surface labelled `OBJ`, the stop surface labelled `STO` and the image surface labelled `IMG`.



	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.

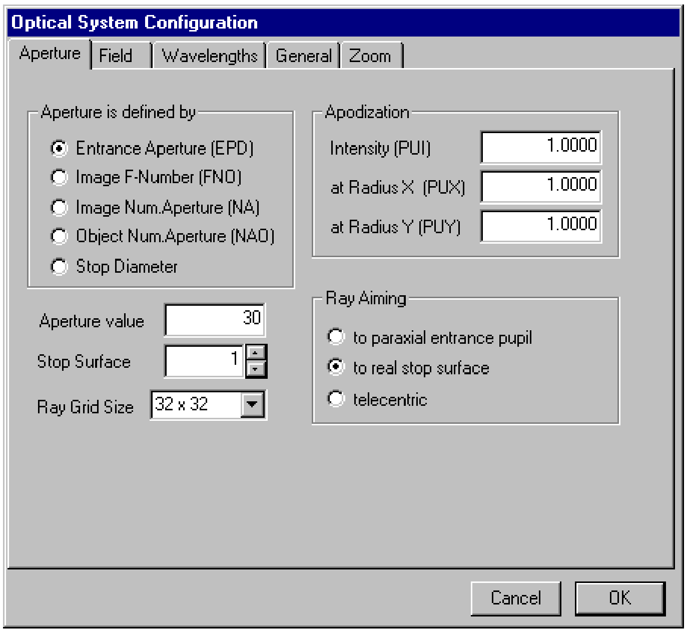
As a minimum, radius of curvature, axial separations and glasses must be entered. For the moment we leave the surface apertures in the APE-Y column zero. They will be automatically calculated later. The surface spreadsheet now should look like this:



	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. 3.7. 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.



Aperture is defined by
● Entrance Aperture (EPD)
○ Image F-Number (FNO)
○ Image Num.Aperture (NA)
○ Object Num.Aperture (NAO)
○ Stop Diameter

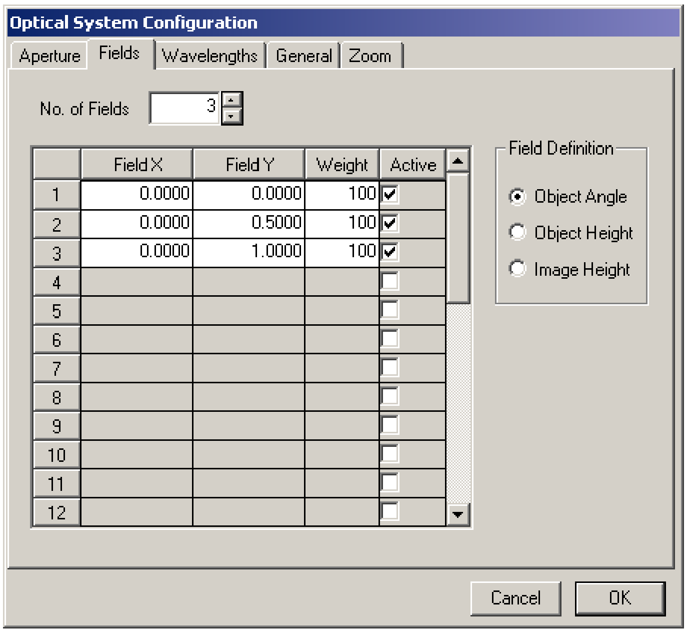
Aperture value 30
Stop Surface 1
Ray Grid Size 32 x 32

Apodization
Intensity (PUI) 1.0000
at Radius X (PUX) 1.0000
at Radius Y (PUY) 1.0000

Ray Aiming
○ to paraxial entrance pupil
● to real stop surface
○ telecentric

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. Note, that the stop surface is at the fourth surface, which is not our preferred location. Although this does not influence the imaging properties very much, since the doublet is used only near to the optical axis, we may want to set the stop surface properly. We set the stop surface to 1. All other settings remain unchanged.

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.

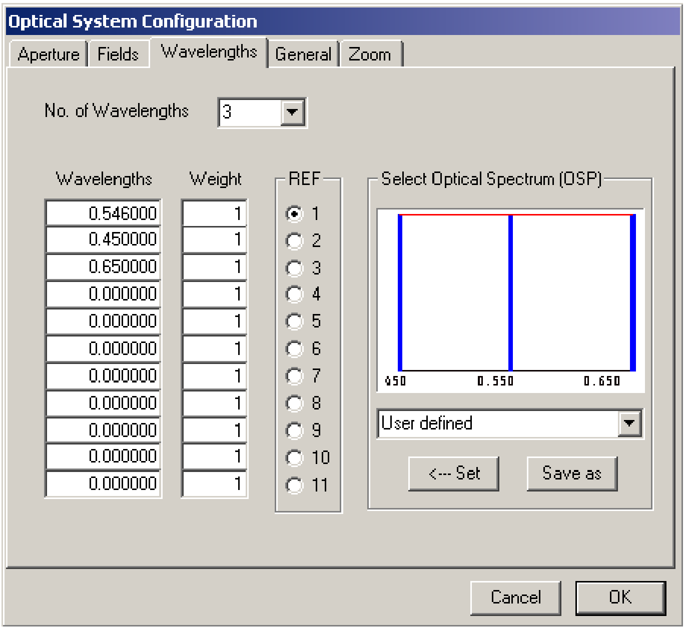


No. of Fields: 3

	Field X	Field Y	Weight	Active
1	0.0000	0.0000	100	☑
2	0.0000	0.5000	100	☑
3	0.0000	1.0000	100	☑
4				□
5				□
6				□
7				□
8				□
9				□
10				□
11				□
12				□

Field Definition
- ● Object Angle
- ○ Object Height
- ○ Image Height

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.



No. of Wavelengths: 3

Wavelengths	Weight	REF
0.546000	1	● 1
0.450000	1	○ 2
0.650000	1	○ 3
0.000000	1	○ 4
0.000000	1	○ 5
0.000000	1	○ 6
0.000000	1	○ 7
0.000000	1	○ 8
0.000000	1	○ 9
0.000000	1	○ 10
0.000000	1	○ 11

Select Optical Spectrum (OSP): User defined

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..4              ! Inserts four 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        ! Defines three wavelengths.
0.65
ref 1                ! Reference wavelength number is 1.
set mht              ! Sets maximum required surface heights according to the speci-
                     fied 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.otx`. 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.

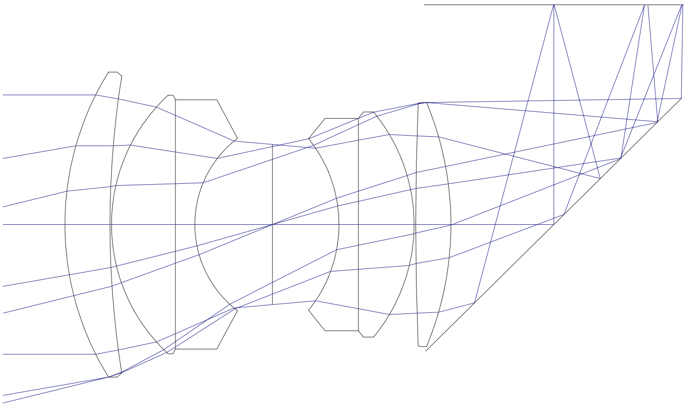


Figure 4.1: Double-Gauss Example with fold mirror.

4.1.1 Menu Entry

The starting design is restored (loaded) from the main menu *File –> Open* and browsing in the file selection box to the `misc` subdirectory. Select the file `double_gauss.otx` and press OK. We may also wish to load the surface spreadsheet editor by selecting *Edit –> Surface data* from the menu or directly clicking on the toolbar icon.

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.

The optical system now has 13 surfaces. 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

The parameters for tilt and position of the mirror surface will now be specified. 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).

Finally, 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 and bends the beam ’up’ after reflection at the mirror surface. As a final step we change the axial distance at surface 11 in the ”Standard data” tab to 10mm in order to get the fold mirror approximately in the middle of last lens surface and image surface. The resulting lens layout plot (`VIE` command) should now be identical to Fig. 4.1.

4.1.2 Command Line Entry

It is assumed that the Double-Gauss example, (found in the examples directory under `examples\misc\double_gauss.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. 4.1 for the result) :

`ins s12`	Inserts a surface before surface 12 (the image surface). The number of the image surface increases to 13.
`sut s12 SDM`	Surface type of surface 12 is spherical (S), decentered (D) and mirror (M).
`ben s12`	Changes tilt mode to `BEN`d (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.
`thi s11 10`	Changes axial thickness at surface 11 to 10mm.
`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. 4.2. 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).

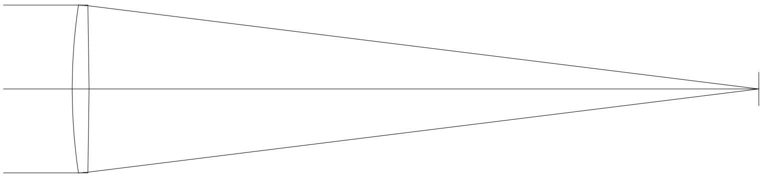


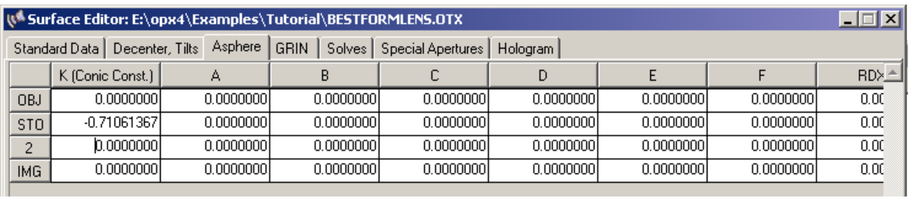
Figure 4.2: Best-form lens taken as staring example for aspherizing a lens.

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 $K = -0.7106137$ must be entered in the column labelled ”K” (the first column) on surface 1 (2^{nd} row), as shown in Fig. 4.3.

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.



Surface Editor: E:\opx4\Examples\Tutorial\BESTFORMLENS.OTX

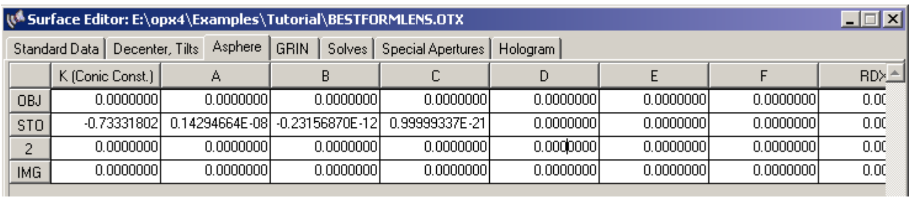
	K (Conic Const.)	A	B	C	D	E	F	RDX
OBJ	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00
STO	-0.71061367	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00
IMG	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00

Figure 4.3: Entering conic constant K.

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. 4.4. Enter the first three coefficients, A through C :

A	0.14294664E-08
B	-0.23156870E-12
C	0.99999337E-21



Surface Editor: E:\opx4\Examples\Tutorial\BESTFORMLENS.OTX

	K (Conic Const.)	A	B	C	D	E	F	RDX
OBJ	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00
STO	-0.73331802	0.14294664E-08	-0.23156870E-12	0.99999337E-21	0.0000000	0.0000000	0.0000000	0.00
2	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00
IMG	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.00

Figure 4.4: 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 -0.7106137
```

Higher order aspheric terms are entered by

```
A s1 0.14294664E-08
B s1 -0.23156870E-12
C s1 0.99999337E-21
```

4.3 Gradient Index (GRIN) Example

Similarly to improving the on-axis performance of a single lens (by asphherizing 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. 4.2 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. 4.5. 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.

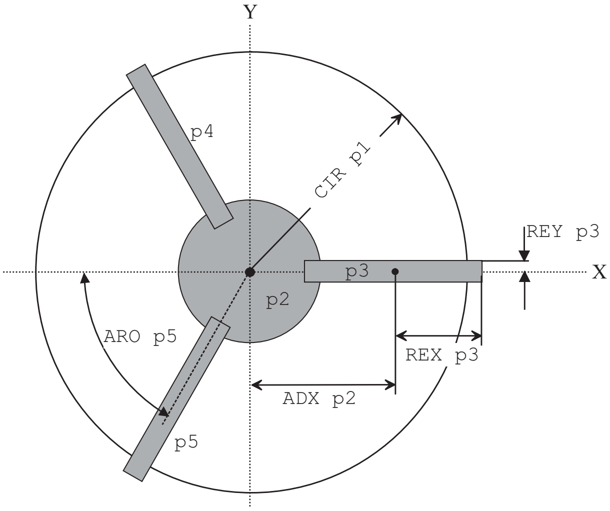


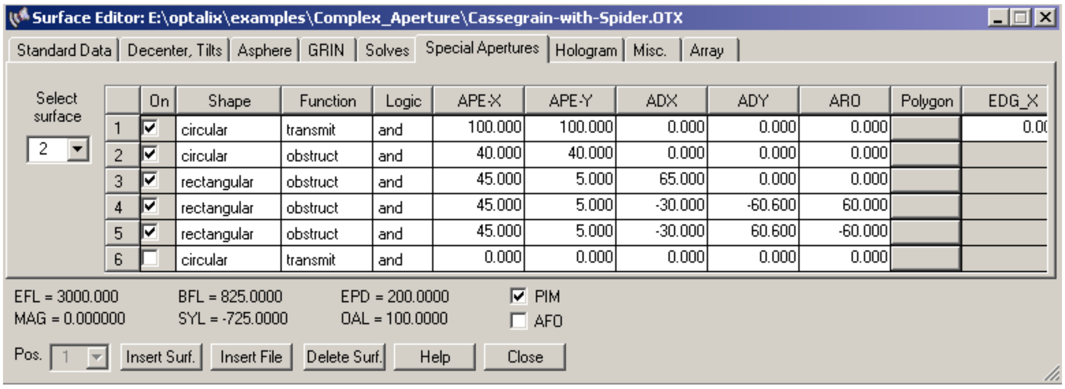
Figure 4.5: Circular aperture with circular central obstruction and spider with 3 vanes.

This aperture is modelled by 5 basic apertures which are logically ANDed. 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. 4.6.



Select surface: 2

	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		

EFL = 3000.000 BFL = 825.0000 EPD = 200.0000 ☑ PIM
MAG = 0.000000 SYL = -725.0000 OAL = 100.0000 ☐ AFO

Pos. 1

Figure 4.6: 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. 4.7 for a sample plot.

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.

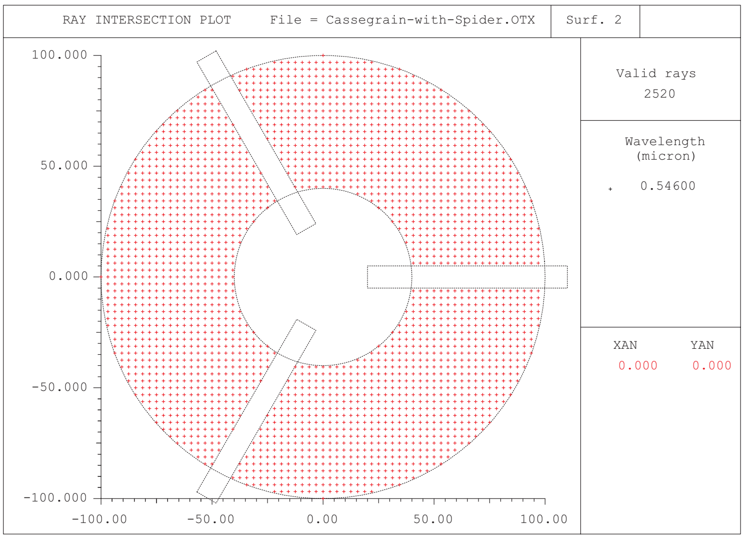


Figure 4.7: Ray intersection plot showing the used areas on a surface with central obstruction and spiders. The necessary apertures to produce this plot are shown in Fig. 4.6. This plot can be generated by restoring the file `$i\examples\complex_aperture\Cassegrain-with-spider.otx` and plotting ray intersections and aperture elements by the command `SPO RIS s2 100`.

4.7 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 60 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.

[1] For more than 60 zoom/multi-configuration positions a text based editor is provided. It is invoked by the ZED command.

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\Laikin-35-1.otx`. This is a 10x mechanically compensated lens taken from Laikin's book "Lens Design" [26]. The optical layout and the movements of the lens groups is shown in Fig. 4.8.

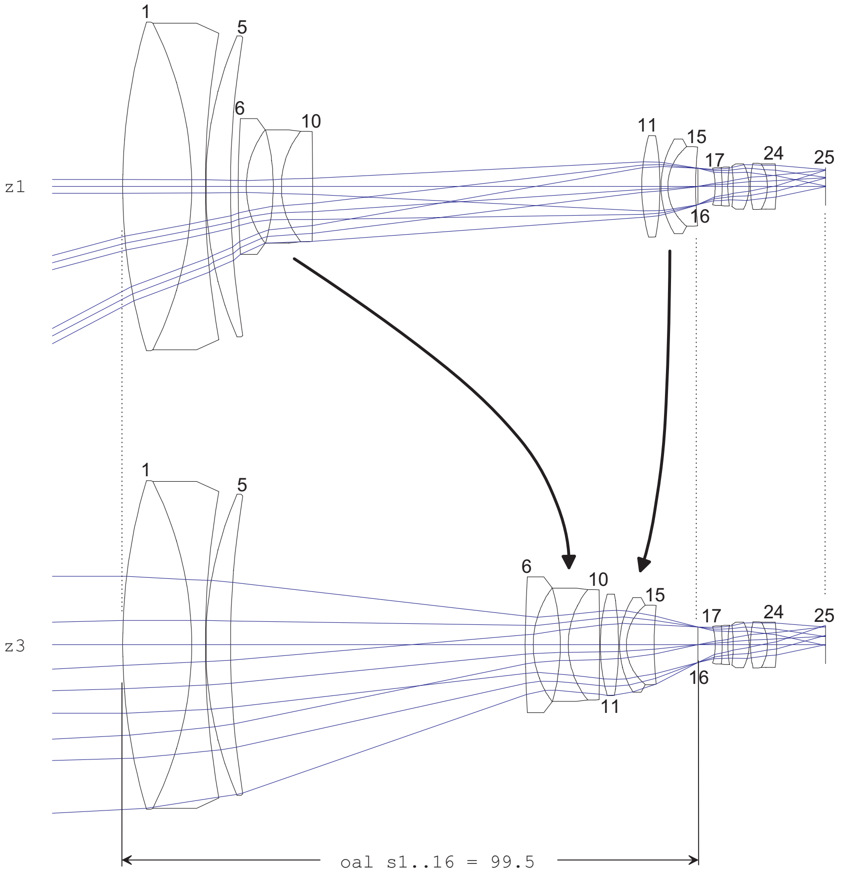


Figure 4.8: Mechanically compensated zoom lens from Laikin [26], example 35-1. Here shown for zoom positions z1 and z3. The middle position (z2) has been left out for sake of clarity.

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.

[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.

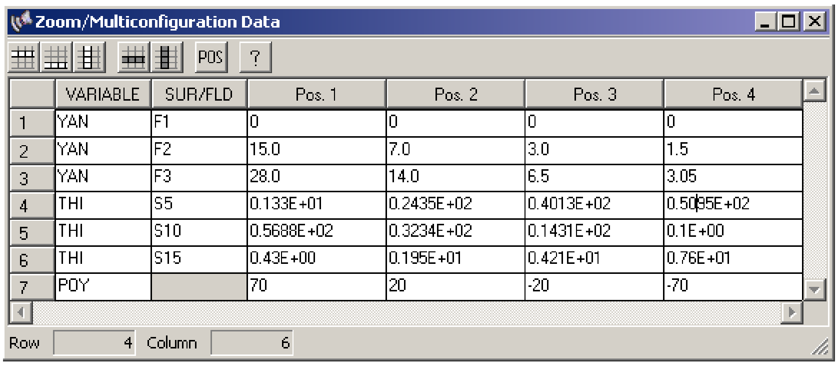
In the command line this is accomplished by the ZOO prefix to each command. For example, the thickness `thi s5` becomes, depending on the zoom position:

```
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. 4.9 shows the zoom setup in the zoom editor (invoked by the command `edi zoo`, or by clicking on the icon ZOO).



	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

Row 4 Column 6

Figure 4.9: 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. 4.8 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. 4.10 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

```
glb s16 1
```

Alternatively, in the surface editor, the global reference is entered in the "Glb" column in the 'General' tab (see Fig. 4.11). 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

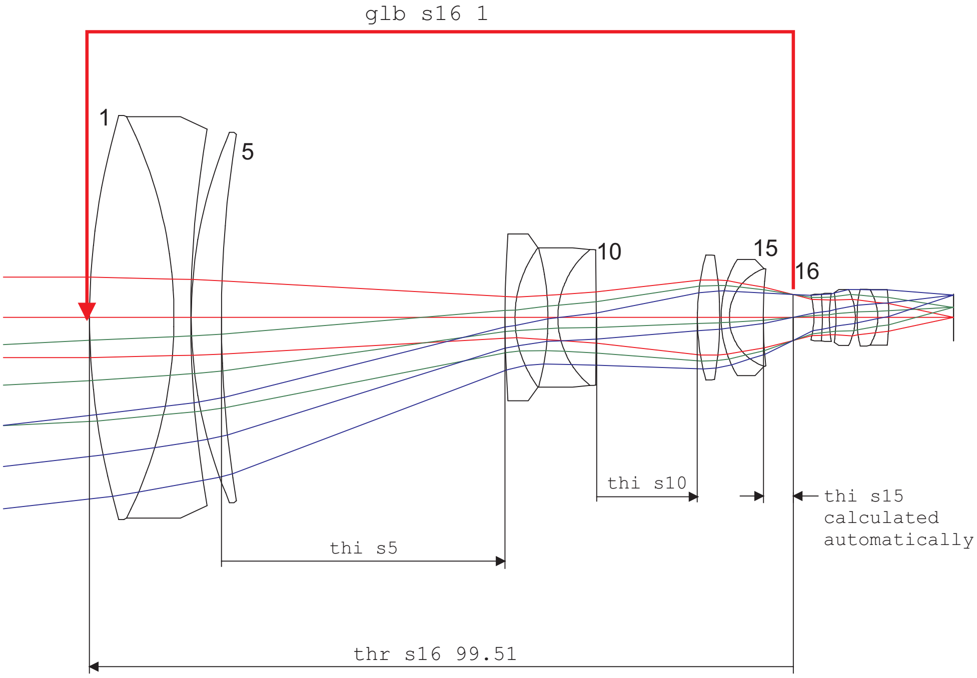
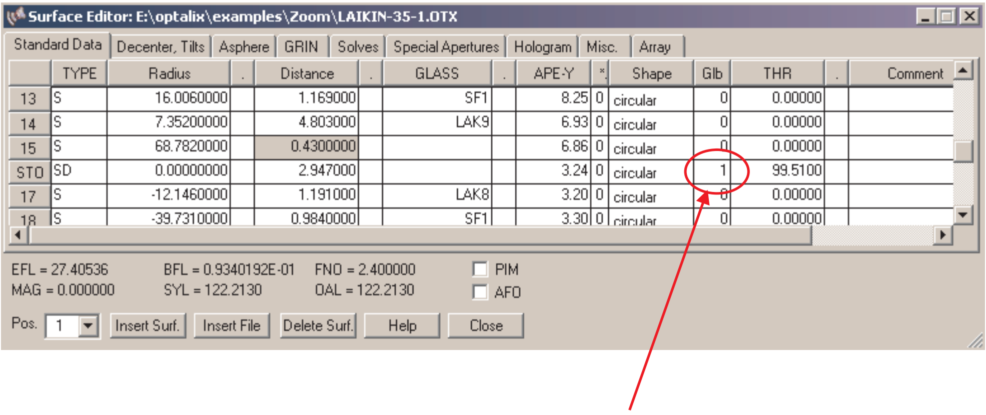


Figure 4.10: Global reference. In this example, the coordinates of surface 16 are referred to surface 1. The reference thickness (THR) is the axial thickness measured ’left’ towards the reference surface.


Surface Editor: E:\optalix\examples\Zoom\LAIKIN-35-1.OTX

	TYPE	Radius	.	Distance	.	GLASS	.	APE-Y	*	Shape	Glb	THR	.	Comment
13	S	16.0060000		1.169000		SF1		8.25	0	circular	0	0.00000		
14	S	7.35200000		4.803000		LAK9		6.93	0	circular	0	0.00000		
15	S	68.7820000		0.4300000				6.86	0	circular	0	0.00000		
STO	SD	0.00000000		2.947000				3.24	0	circular	1	99.5100		
17	S	-12.1460000		1.191000		LAK8		3.20	0	circular	0	0.00000		
18	S	-39.7310000		0.9840000		SF1		3.30	0	circular	0	0.00000		

EFL = 27.40536 BFL = 0.9340192E-01 FNO = 2.400000 □ PIM
MAG = 0.000000 SYL = 122.2130 OAL = 122.2130 □ AFO

Pos. 1

Global reference to surface 1

Figure 4.11: Defining global reference in the surface editor. In this example, the coordinates of surface 16 are referred to surface 1.

(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 k with respect to a prior surface.

5

Special Options

5.1 Macros

5.2 User Defined Graphics (UDG)

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

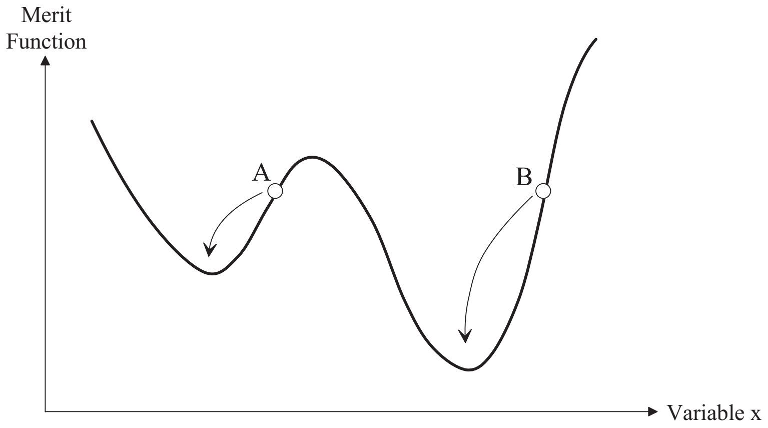


Figure 6.1: Merit function for one variable. The choice of the starting point (A, B) determines which local minimum is found.

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

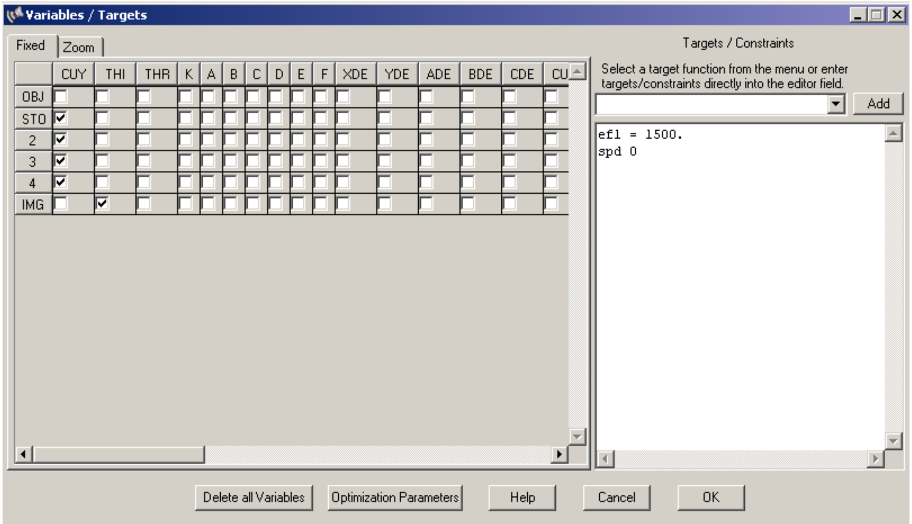
Construction parameters are made variable by the command "`VAR`" or by clicking on the VAR icon. A dialog box as shown in Fig. 6.2 will be opened. In our Buchroeder design, we will make all curvatures variable, that is `CUY` (curvature Y) at surfaces 1 to 4. Thus, we check surfaces 1 to 4 in the CUY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

In addition, we will also make the defocus (DEF) variable. The defocus is an additive value to the paraxial focus. It helps finding the minimum spot diameter, which generally is not at the paraxial focus, mainly due to residual spherical aberration. We check the box labelled `DEF` in the lower left corner of this dialog box.

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, 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



	CUY	THI	THR	K	A	B	C	D	E	F	XDE	YDE	ADE	BDE	CDE	CU
OBJ	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
STO	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
2	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
3	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
4	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
IMG	☐	☑	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Select a target function from the menu or enter targets/constraints directly into the editor field.

```
efl = 1500.
spd 0
```

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

```
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

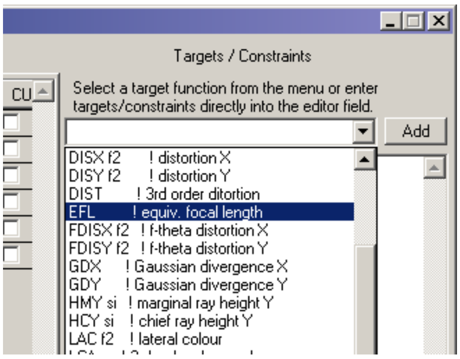
```
EFL =<> target
```

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
```

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


Targets / Constraints
CU
Add
DISX f2 ! distortion X
DISY f2 ! distortion Y
DIST ! 3rd order ditortion
EFL ! equiv. focal length
FDISX f2 ! f-theta distortion X
FDISY f2 ! f-theta distortion Y
GDX ! Gaussian divergence X
GDY ! Gaussian divergence Y
HMY si ! marginal ray height Y
HCY si ! chief ray height Y
LAC f2 ! lateral colour

Select a target function from the menu or enter targets/constraints directly into the editor field.

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

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:

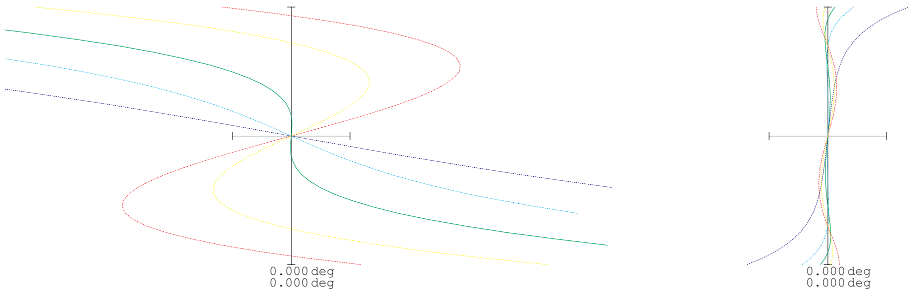
```
Iter          Min.          Equal.          Inequal.          DumpingF.       Improv.
   0      0.906432        6.485413        0.000000         1.000000
   1      0.268796        1.105826        0.000000       0.8133397E-02     0.70346
   2      0.041503        0.102830        0.000000       0.8133397E-03     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 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

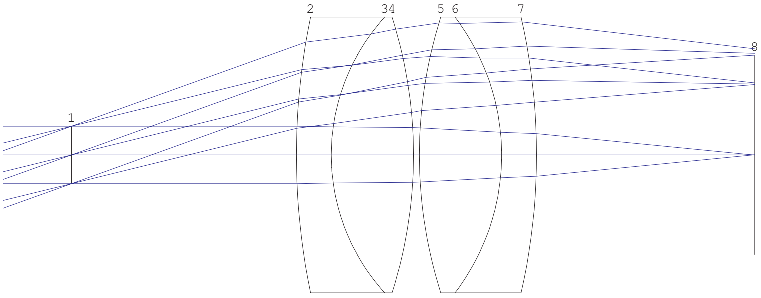


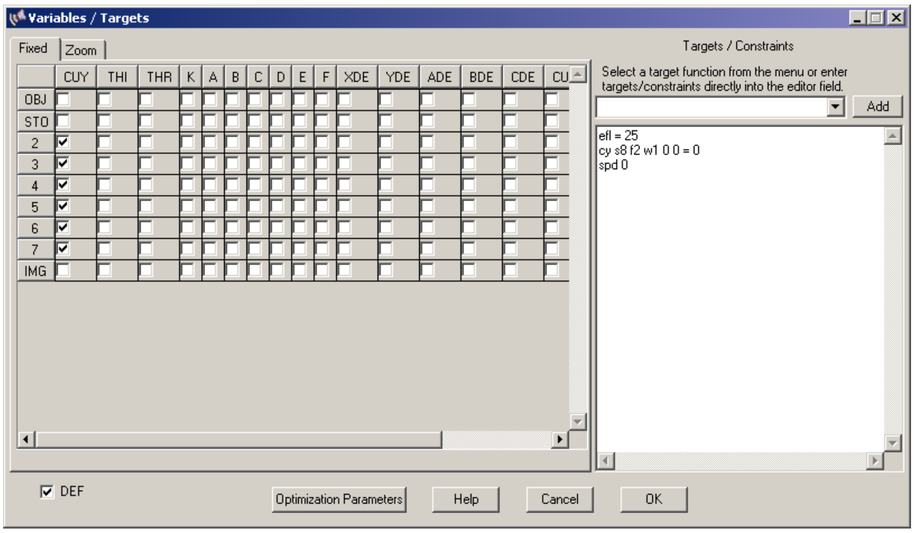
Figure 6.5: Ploessl eyepiece.

(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 VAR command or by clicking on the VAR icon).



	CUY	THI	THR	K	A	B	C	D	E	F	XDE	YDE	ADE	BDE	CDE	CU
OBJ	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
STO	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
3	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
4	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
5	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
6	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
7	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
IMG	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

Select a target function from the menu or enter targets/constraints directly into the editor field.

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

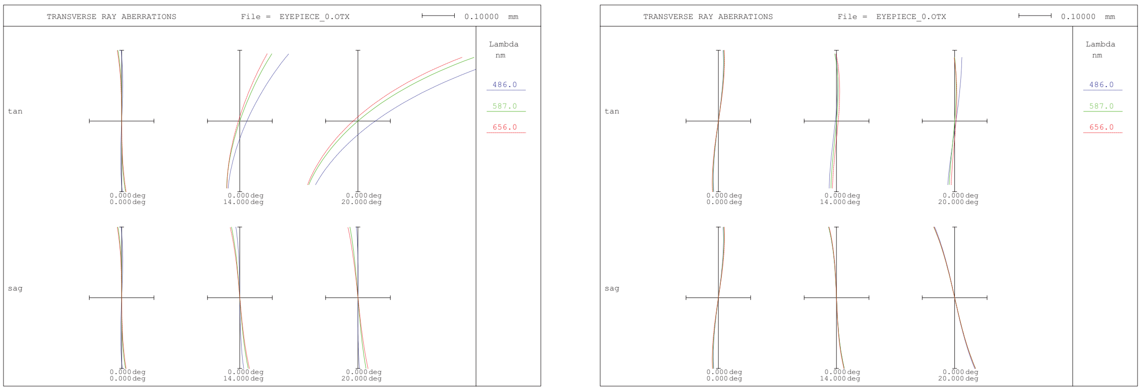
☑ DEF

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

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 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.

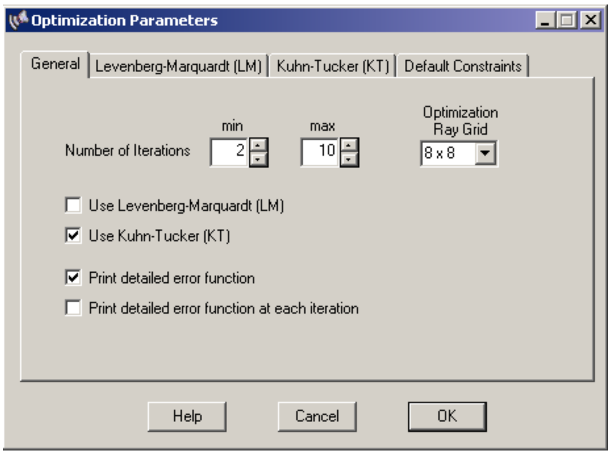
This section provides complementary information for adjusting the optimization process. Since we have already predefined reasonable optimization parameters in all optimization examples found in this tutorial there is no immediate need to alter these values. The reader may skip this section and return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

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).



Number of Iterations min 2 max 10

Optimization Ray Grid 8 x 8

□ Use Levenberg-Marquardt (LM)

☑ Use Kuhn-Tucker (KT)

☑ Print detailed error function

□ Print detailed error function at each iteration

Figure 6.8: Dialog box for editing general optimization parameters.

$OpTaliX$ 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. Consult the reference manual for a detailed description of both methods.

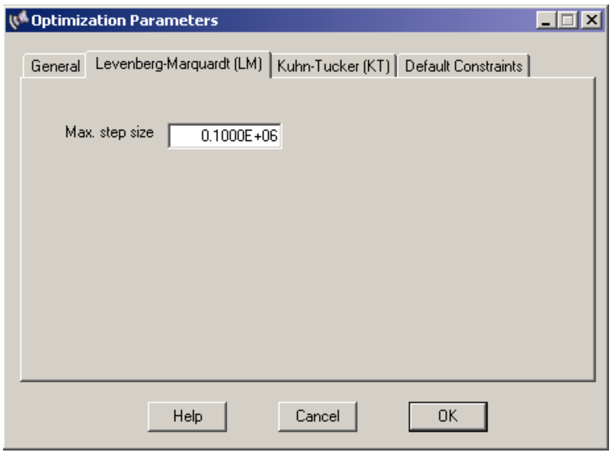
By default, the KT-optimization is used. This setting should not be changed unless otherwise recommended.

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.

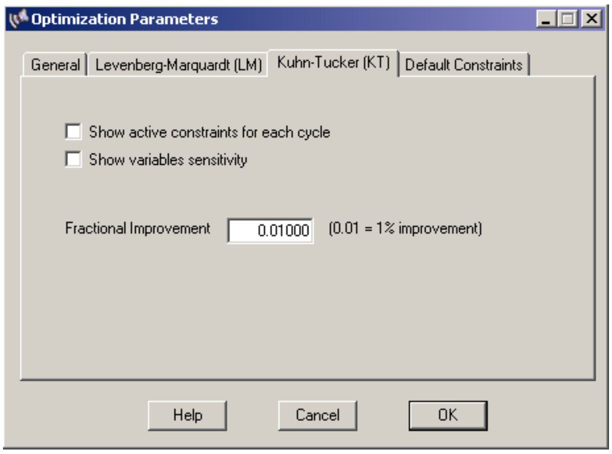
6.1.5.2 Levenberg-Marquardt (LM) Tab

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.



Max. step size 0.1000E+06

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



□ Show active constraints for each cycle
□ Show variables sensitivity

Fractional Improvement 0.01000 (0.01 = 1% improvement)

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

6.1.5.3 Kuhn-Tucker (KT) Tab

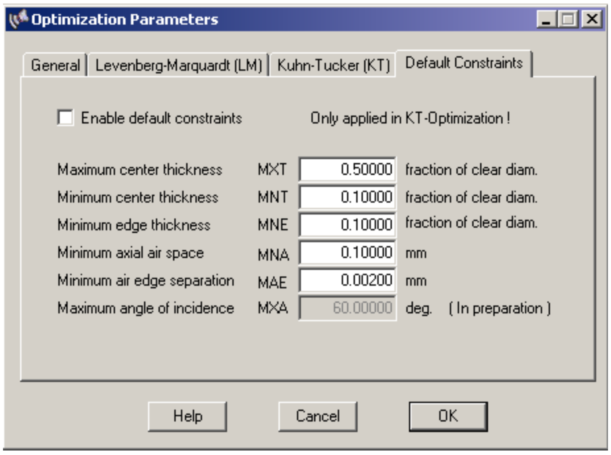
Show active constraints for each cycle: Outputs which constraints are currently active. Inequality constraints are dynamically activated or released as the optimization evolves.

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.

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



□ Enable default constraints   Only applied in KT-Optimization !

Maximum center thickness	MXT	0.50000	fraction of clear diam.
Minimum center thickness	MNT	0.10000	fraction of clear diam.
Minimum edge thickness	MNE	0.10000	fraction of clear diam.
Minimum axial air space	MNA	0.10000	mm
Minimum air edge separation	MAE	0.00200	mm
Maximum angle of incidence	MXA	60.00000	deg. (In preparation)

Figure 6.11: Dialog box for editing default constraints.

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 42). 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_0.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.

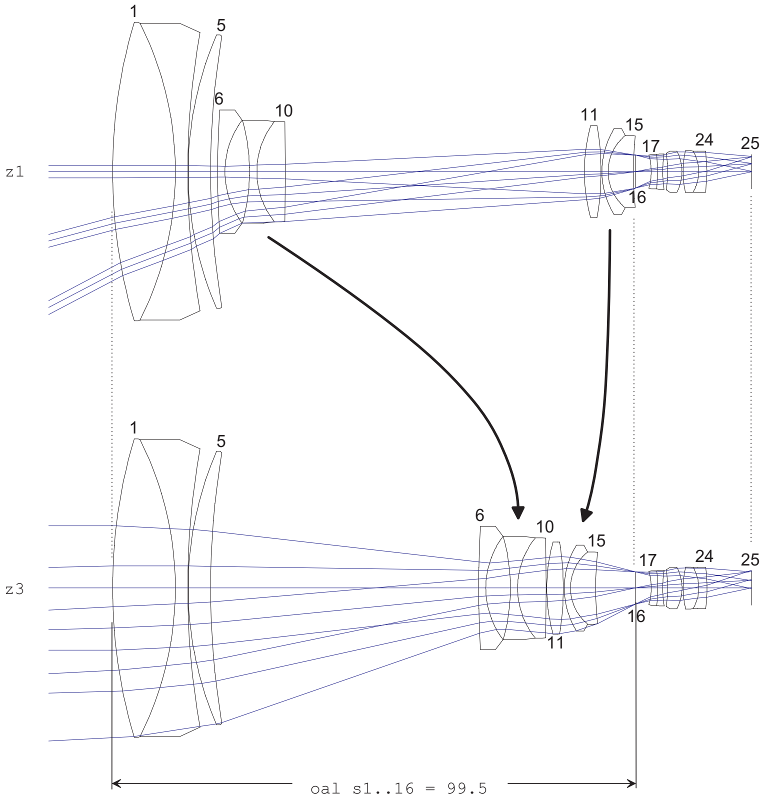


Figure 6.12: Mechanically compensated zoom lens shown for two positions, z1, z3. The middle position z2 has been left out for the sake of clarity.

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

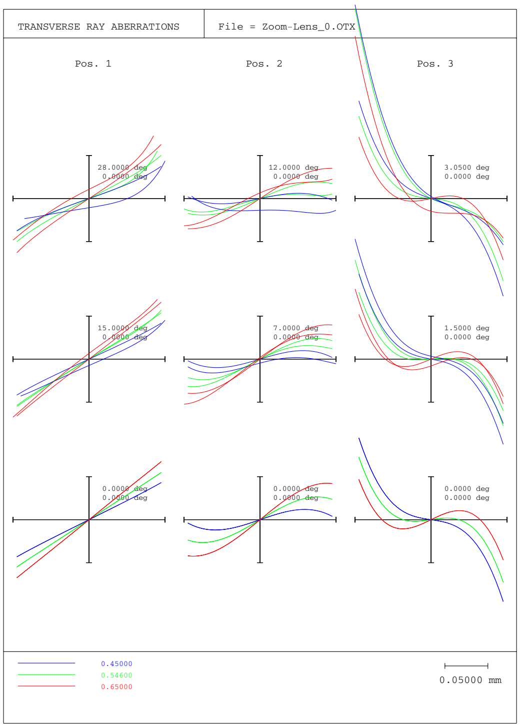


Figure 6.13: Transverse aberrations of the start system (before optimization). Note: If the appearance of your transverse aberration plot is different from this figure, one of your settings in the preferences section will be different. From the main menu *File* $--$ > *Preferences*, select the 'operations'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

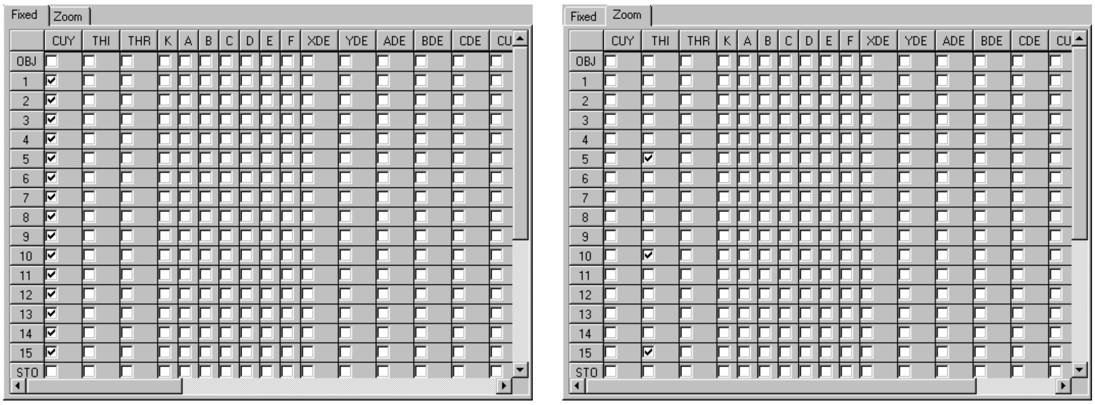
position. We will denote the first kind of variables as "fixed" variables[1] (or non-zoomed variables) and the latter as "zoom(ed)" variables. Clearly the radii of curvatures must remain the same for all positions. Among the thicknesses, three air spaces will vary for each position. These are `THI s5`, `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 VAR icon or by entering the command "`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

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



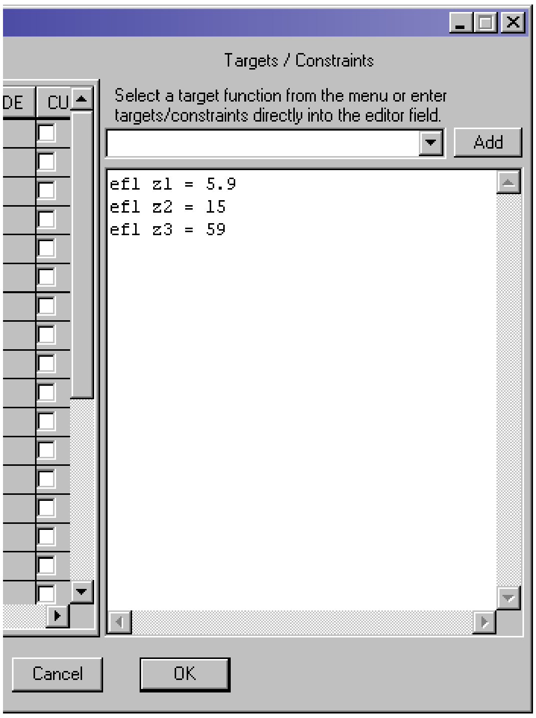
	CUY	THI	THR	K	A	B	C	D	E	F	XDE	YDE	ADE	BDE	CDE	CU
OBJ	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
3	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
4	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
5	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
6	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
7	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
8	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
10	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
11	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
12	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
13	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
14	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
15	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
STO	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

	CUY	THI	THR	K	A	B	C	D	E	F	XDE	YDE	ADE	BDE	CDE	CU
OBJ	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
1	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
2	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
3	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
4	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
5	□	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□
6	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
7	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
8	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
9	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
10	□	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□
11	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
12	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
13	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
14	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□
15	□	☑	□	□	□	□	□	□	□	□	□	□	□	□	□	□
STO	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□	□

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

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 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.



Targets / Constraints

Select a target function from the menu or enter targets/constraints directly into the editor field.

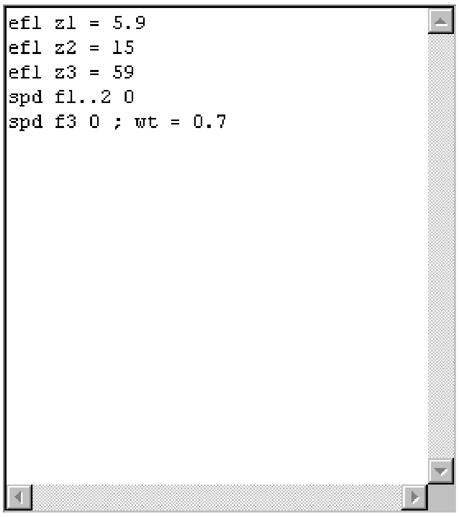
```
efl z1 = 5.9
efl z2 = 15
efl z3 = 59
```

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.



```
efl z1 = 5.9
efl z2 = 15
efl z3 = 59
spd f1..2 0
spd f3 0 ; wt = 0.7
```

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

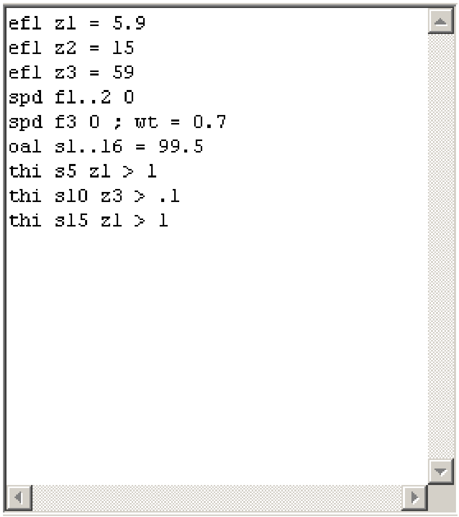
We are now at a point where optimization already could be run, however, the result would be unsatisfactory. Typically a zoom lens should maintain focus during ”zooming” and we also need some additional constraints to keep the first and last group stationary. The latter requirement is fulfilled by requiring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



```
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.

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 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. Type on the command line `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.

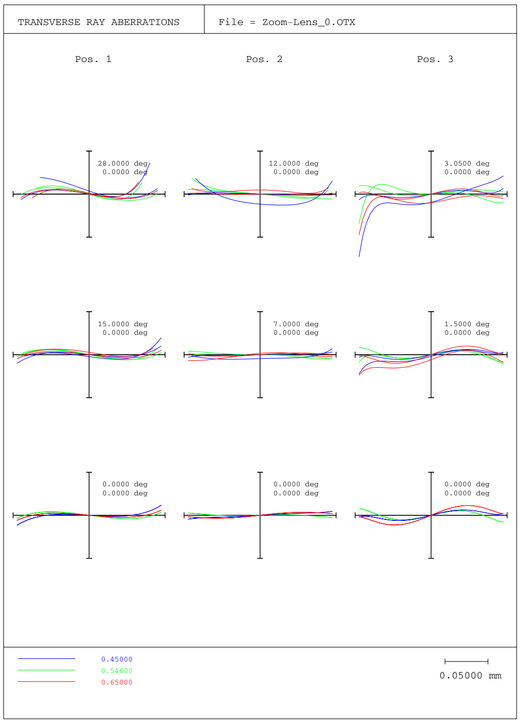


Figure 6.18: Transverse ray aberrations after 10 optimization cycles. Compare also with Fig. 6.13

7

Coatings

7.1 Basics

7.2 Formula Entry

7.3 Coating Analysis

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

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

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

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

[13] 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)

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[35] 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)

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

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

[38] 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

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

[40] 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

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

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

[43] 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

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

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

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

[47] 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

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

[49] 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

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

[51] W.T.Welford, ”A vector raytracing equation for hologram lenses of arbitrary shape”. Optics Communications, Vol.14, No. 3, July 1975, p.322

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

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

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

Index